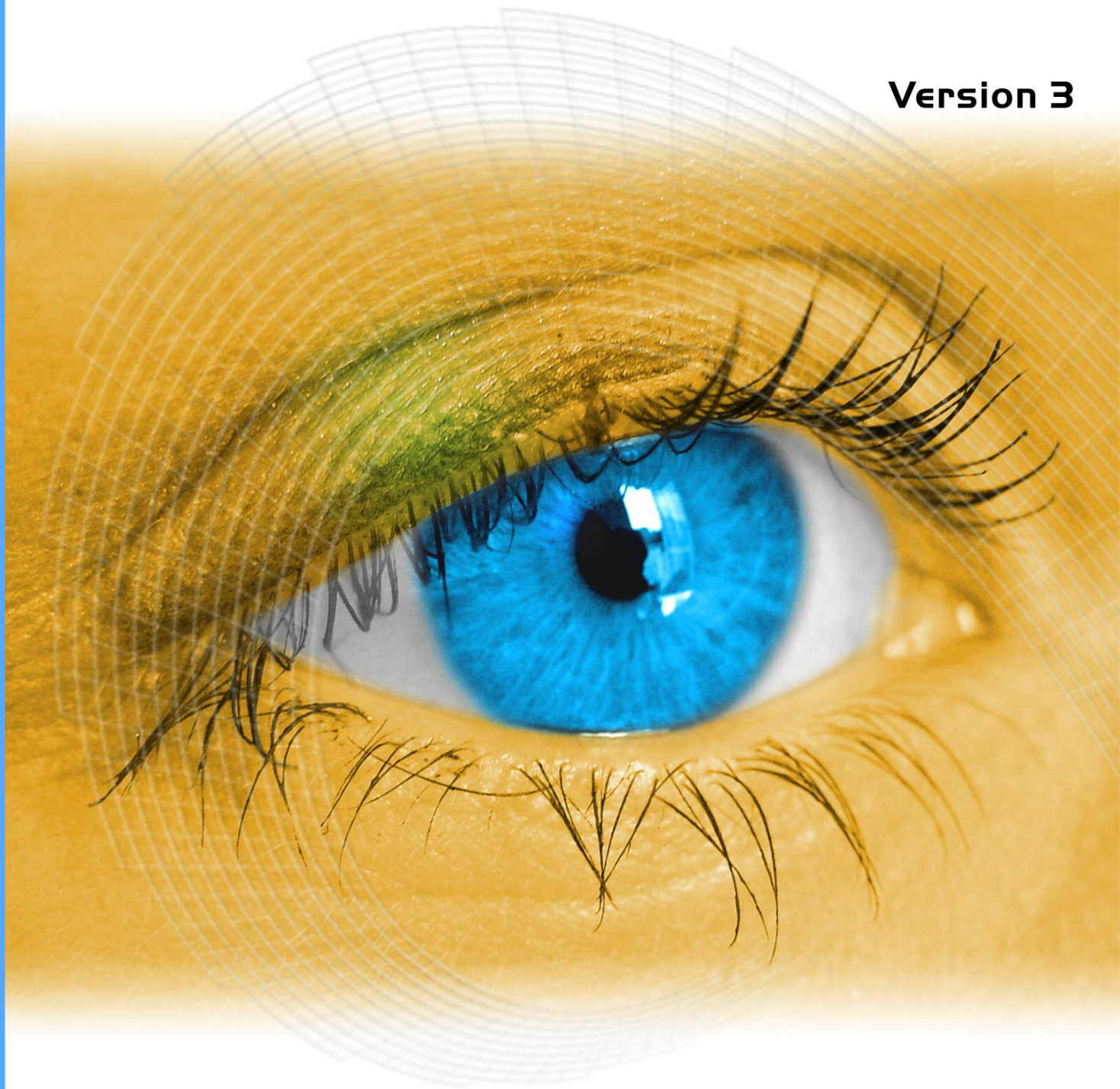
Neptune Labs

FSI Viewer

Flash based Single Source Image Viewer

Version 3



User Manual

FSI Viewer
FSI Showcase Add-on
FSI Plug-ins

Software:
NeptuneLabs GbR
P.O.B. 1207
32818 Blomberg
Germany

© 2003-2005 NeptuneLabs GbR. All rights reserved.

Last updated in April 2005
FSI Viewer Version 3.2.5
FSI Showcase Version 3.1.2
Manual Revision 019

All brands and product names are trademarks or registered trademarks of the respective producers.

Contents

Contents 5
Alphabetical Index of FSI Viewer Parameters 7

INTRODUCTION 9
Compatibility and System Requirements 10
Using FSI Viewer 11
Menu Bar 11
Mouse Modes 12
Additional Buttons 13
Survey Window 13
Keyboard Shortcuts 14

SETUP 15
Setting up FSI Viewer 16
Licensing 17

HTML-SOURCE CODE 18
Integration into HTML Pages 18
Object Tag 18
Embed Tag 18
Variable HTML Parameters 19

CONFIGURATION 20
Using XML Configuration Files (*.fsi) 20
Using HTTP Queries 22
Hierarchy of Configuration Parameters 23
Choosing a Configuration Method 24
Retrieving Parameters Automatically 25
Basic Parameters 27
Optional Parameters 30

VIRTUAL 3D PRESENTATIONS 47
3D Parameters 49

DEBUG MODE 53
Enabling the Debug Mode 53
Using the Debug Window 53

AUTOMATED IMPLEMENTATION OF IMAGES 55

PLUG-INS 58

Using Plug-ins 58
Deactivating Plug-ins 59
Plug-in Parameters 60
Providing Plug-in Parameters by Query 60

FSI SHOWCASE ADD-ON 61
Integrating FSI Showcase 61
Defining Image Lists 61
Defining the Showcase Layout 64
Using FSI Showcase and FSI plug-ins 66
Showcase Parameters 67
Alphabetical Index of FSI Showcase Parameters 68
Basic Parameters 69
ImageList Parameters 74
Advanced Parameters 76

PLUG-IN REFERENCE 86
Plug-in Mousemodes 86
Plug-in Mousemode Select 88
Plug-in Zoom Meter 90
Plug-in Max Zoom 91
Plug-in History 92
Plug-in Resize 94
Plug-in CustomButton 96
Plug-in NavExtension 100
Plug-in Software Cursor 102
Plug-in MouseWheel 103
Plug-in ClockProgress 106
Plug-in Music 109
Plug-in Magnifier 111
Plug-in HotSpots 113
Plug-in JavaScript Bridge 123
Plug-in Measure 132
Plug-in SelectFrame 138

ERROR MESSAGES 142

APPENDIX 145

INDEX 149

Alphabetical Index of FSI Viewer Parameters

Animation 39
AnimationSpeed 39
Cfg 27
ConcurrentRequests 45
Debug 30
DefaultCfg 27
DetailBuffer 33
Effects 40
ExtendedViewport 39
FPXBase 31
FPXHeight 29
FPXServerTemplate 28
FPXServerType 28
FPXSrc 28
FPXTilePaddingX 51
FPXTilePaddingY 51
FPXTilesX 49
FPXTilesY 49
FPXWidth 29
FSIBase 31
HandCursor 38
HelpUrl 37
HelpURLTarget 37
HideUI 40
IgnoreQuery 44
InitialAction 36
InitialActionDelay 36
InitialActionPersistent 37
InitialMouseMode 37
InitialView 35
InitialViewPersistent 36
Intro 30
Language 33
MaxZoomLevel 44
MenuAlign 33
NavEffects 43
NavFrame 34
NavFrameColor 34
NavHeight 34
NavMaskAlpha 35
NavMaskColor 35
NavWidth 34
NoImageBlend 40
NoNav 34
NoSceneAnimation 51
NoSceneLoop 52
NoSetLoop 51
SceneEffects 43
ScenePreload 51
SceneSets 50
Skin 30
TileEffects 43
TileSizeX 45
TileSizeY 45
UISwitchable 40
ViewerBorder 32
ViewerHeight 32
ViewerWidth 32
ZoomCache 43
ZoomCache5Only 44
ZoomCacheID 44
ZoomLevelFrequency 46
ZoomLimit 38
ZoomLimitMin 38

NeptuneLabs FSI Viewer
Introduction

FSI Viewer ('**F**lash based **S**ingle Source **I**mage Viewer') and FSI Showcase Add-on have been designed to display high resolution images on the internet requiring low bandwidth only. Using the Macromedia Flash™ browser plug-in FSI Viewer requests image data from Single Source Imaging Servers.

Using FSI Viewer you can present two-dimensional images as well as three-dimensional presentations consisting of multiple individual images. By using Single Source Imaging Servers only one high resolution source image is required for each image to be displayed.

By using the Macromedia Flash™ plug-in FSI Viewer and FSI Showcase Add-on can be integrated almost independently of the browser version. As today the penetration of this plug-in exceeds 95% of all internet users FSI products can usually be used without downloading or setting up a browser plug-in.

By means of a large number of configuration options and optional FSI Skins, FSI Viewer and FSI Showcase Add-on can widely be adjusted to integrate seamlessly into your website. Using the FSI Plug-in system you can integrate additional functionality into FSI Viewer at runtime without increasing the download size in general.

Thank you for using NeptuneLabs software!

www.fsi-viewer.com - Online Resources for FSI Viewer

Please be sure to visit www.fsi-viewer.com for software updates, regularly updated samples, tutorials and downloads. To access restricted areas on the website you can use the username and password that came with your licensed copy of FSI Viewer.

Compatibility and System Requirements

Imaging Servers
FSI products covered by this manual can at present be securely run with eRez Imaging Server from version 2.54, iSeeMedia Zoom Image Server from version 4 and TrueSpectra version 4.1.1.

Macromedia Flash™
FSI Viewer requires a Flash™ Plug-in version 5 or newer for the display in browsers. If you are using Flash™ 5, NeptuneLabs ZoomCache is required to output the zoom images. This does not apply to eRez Imaging Server, which supports Flash™ 5 plug-ins without ZoomCache.

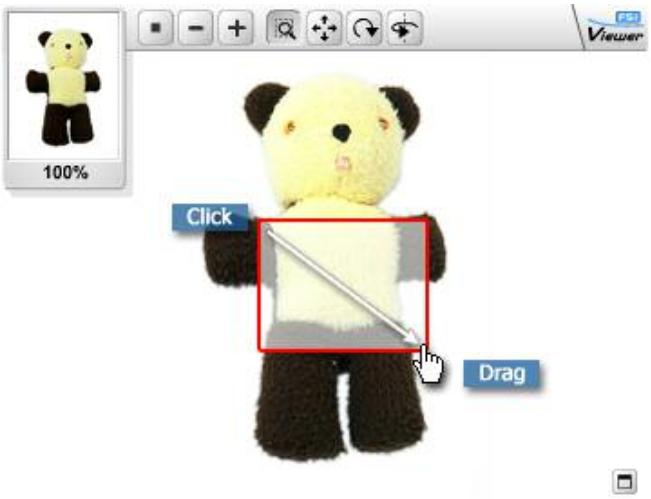
To run FSI Showcase Add-on, Flash™ version 6 or newer is required.

FSI Viewer has been successfully tested with the following software versions:

YaWah eRez	-	Imaging Server 2.5.4 – 3.1.0
iSeeMedia	-	ZoomImageServer 4.0 – 4.6
TrueSpectra	-	Image Server 4.1.1 Service Patch 3
Flash™-Plug-in	-	5.0.42.0 to 7.0.19.0

The presentation of FSI Viewer may marginally vary depending on the operating system. If problems arise please check the versions mentioned above and enquire about newer versions of the FSI products if necessary.

Using FSI Viewer

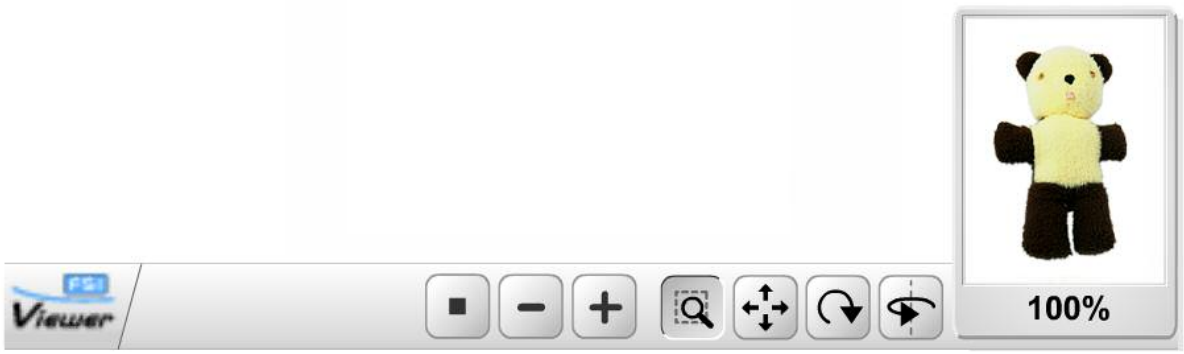


FSI Viewer is navigated by the menu bar (above) and by using the mouse directly on the image. For example, you can directly select a section of the image you want to magnify.

The mouse functions are determined by the corresponding buttons on the menu bar (zoom, pan, rotate, etc.)
The example to the left shows the mouse mode "zoom".

The optional small survey window (above left) displays the position of the image section currently viewed.

Menu Bar



The menu bar might look different depending on skins or additional plug-ins.

Main Functions

	Back to Initial View. Undoes zoom, pan, and rotation and restores the initial view. (identical with space bar)
	Zoom In
	Zoom Out

Mouse Modes

The "Mousemodes" or "MousemodeSelect" plug-in is required to display the following mouse mode buttons.

	Mouse - Zoom In Choose this function to enlarge segments using the mouse. Click on the image and drag the frame over the desired segment. Alternatively you can click on the image, without marking a segment. The image will then be magnified in steps. To zoom out in steps, hold down the CTRL-key and click on the image.
	Mouse - Pan Choose this functions to pan the image using the mouse. Click on the image and drag in the desired direction. To return to the original view, hold down the CTRL-key and click on the image.
	Mouse - Turn Choose this function to rotate the image around the z-axis. Click on the image and drag in whatever direction you want to turn the object (to the right or the left). To reset the rotation, hold down the CTRL-key and click on the image.
	Mouse - Rotate (only for 3D presentations) Choose this function to rotate the object around the y-axis or the x-axis if available. Click on the image and drag to the left or to the right. Move the mouse up or down to rotate the object around the x-axis. To reset rotation, hold down the CTRL-key and click on the image.

Additional Buttons

	Hide/Display menu Displays or hides the user interface.
	Information Displays information about the viewer and refers to a configurable help page (→Parameter HelpURL).

Survey Window

In the survey window you can see a miniature presentation of the entire image. The segment which is currently viewed is framed in red. In the survey window you can change the current segment either by dragging the frame, or by clicking on the desired area of the image.

Keyboard Shortcuts

FSI Viewer and FSI Showcase can also be controlled with the keyboard. The list below shows which keys have what function. As most of the keys belong to the Num field, it has to be activated before use.

Key	Function
Num 5, spacebar	Back to original view
Num 4, left arrow	Move to the left
Num 6, right arrow	Move to the right
Num 8, up arrow	Move up
Num 2, down arrow	Move down
Num 1	Turn around the z-axis to the left
Num 3	Turn around the z-axis to the right
Num 7	Rotate around the y-axis to the left (3D only)
Num 9	Rotate around the y-axis to the right (3D only)
Num /	Rotate around the x-axis upwards (3D only), or previous view
Num *	Rotate around the x-axis downwards (3D only), or previous view
Num 0	Hide/display user interface
D	--- If you have activated the debug mode, you can hide or display the debug window using this key.
I	--- Outputs information about the current image segment and further status information in the debug window. (e.g. for InitalView).

NeptuneLabs FSI Viewer
Setup

Setting up FSI Viewer is as easy as copying all FSI Viewer files to your web server. The only file that will directly be accessed by the web browser is **fsi.swf** located in the main index.
As the URL to this file is part of each HTML code presenting FSI Viewer it is recommended to keep the path to fsi.swf as short as possible by either:

a) Creating a subdirectory for FSI Viewer in the root directory of you web server. This way the file can easily be addressed relatively from any location of your website.

Example – HTML code addressing FSI Viewer in root directory

```
<PARAM name="Movie" VALUE="/fsi/fsi.swf?cfg=foo">
```

b) Creating a sub-domain to access FSI Viewer absolutely.

Example – HTML code addressing FSI Viewer by subdomain

```
<PARAM name="Movie" VALUE="http://fsi.domain.com/fsi.swf?cfg=foo">
```

Index structure of FSI Viewer

Directory	Content Description
core	Core components
debug	Debugging components *This folder can safely be deleted if debugging is not required any more*
languages	User interface language files
plugins	FSI Plug-ins extending FSI Viewer at runtime
skins	FSI Skins containing the visible parts of the user interface

Setting up FSI Viewer

Step 1

Copy the FSI Viewer and/or FSI Showcase Add-on files using an FTP program into the corresponding directory on your web server.
Please note that all files have to be transferred in binary mode.

Step 2

Edit the default configuration file "**_default.fsi** " located in the main index of your setup directory using a text editor of your choice.
The most important modifications are:

- → **FPXServerType**
  Enter the type of your imaging server, e.g. "erez" or "ZIS"
- → **FPXBase**
  Request base of the imaging server, e.g.

  "`http://domain/erez3/erez?src=`"
  or
  "`http://zis.domain.xcom/fif=`"

Additionally you can create configuration files for individual images or use HTTP queries to access different images. Please refer to the chapter →"**Automated Implementation of Images**" if you plan to integrate a large number of images.

Step 3

Add the HTML source for a Flash movie to your website and enter the path to the file "fsi.swf" with the desired parameters as a HTTP query for the parameter "Movie" (object tag) and "Src" (embed tag).

```
<PARAM NAME="movie" VALUE="fsi.swf?cfg=foo&debug=1">
```

Please use the files in the "**samples**" folder of the installation archive as a guide line.

Licensing

You can use FSI Viewer without a valid licence key for evaluation purposes. In this case the phrase "Evaluation Copy" appears on top of the image and the about box displays "Unregistered Evaluation Copy" in the "Licensed to" section.

After purchasing an FSI Viewer licence you will receive a licence key that removes the "Evaluation Copy" mark and your name appears in the "licensed to" section of the about box.

Entering the Licence Key

After receiving your licence key you need to add the key to your **_default.fsi** file located in the main index of your FSI Viewer setup directory.

1. Open the file "**_default.fsi**" using a text editor of your choice
2. Copy the licence key you received
3. Paste the licence key (encapsulated in a <Licence> tag)
4. Save the modified file

Example – Licence key added to the _default.fsi file

```
<fsi_parameter>

   <licence>
      [Licence key goes here]
   </licence>

   ...
</fsi_parameter>
```

Make sure to flush your browser cache as the _default.fsi file might have been cached by your browser.

Please note that setting up FSI Viewer on additional domains or using unregistered FSI Plug-ins or FSI Skins might activate the evaluation mode despite of the licence key you entered.

NeptuneLabs FSI Viewer HTML-Source Code

Integration into HTML Pages

NeptuneLabs FSI Viewer has to be integrated into HTML pages by means of an <object> - tag, just like any other Flash movie clip. Configuration parameters have to be appended to the URL of the movie clip be means of an HTTP query. If you already know how to add Flash movie clips to HTML code you can simply skip this chapter and move on to the chapter "Configuration".

Object Tag

All recent browsers require an <Object> tag to integrate a Flash movie clip. Older browsers use the obsolete <Embed> tag instead.

Sample HTML code to integrate FSI Viewer into HTML pages:
(Variable data printed in bold and square brackets):

Example - HTML integration without considering old browsers

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfl
ash.cab#version=5,0,42,0" width="[Width]" height="[Height]">
<PARAM NAME="movie" VALUE="[URL and Parameter]">
<PARAM NAME="bgcolor" VALUE="[Background color]">
<PARAM NAME="menu" VALUE="false">
</OBJECT>
```

Embed Tag

For browsers (e.g. Netscape 4.x) that do not support the <object>-tag, an additional <embed>-tag with identical parameters is required.

Example - HTML integrations for modern and old browsers

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfl
ash.cab#version=5,0,42,0" width="[Width]" height="[Height]">
<PARAM NAME="movie" VALUE="[URL and Parameter]">
<PARAM NAME="bgcolor" VALUE="[Background color]">
<PARAM NAME="menu" VALUE="false">
<EMBED TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1
_Prod_Version=ShockwaveFlash" SRC="[URL and Parameter]"
WIDTH="[Width]" HEIGHT="[Height]" BGCOLOR="[Background color]"
MENU="false"></EMBED>
</OBJECT>
```

Variable HTML Parameters

[Width]	Width of the viewer in pixel (or optional in percent with flash 6 or newer)
[Height]	Height of the viewer in pixel (or optional in percent with flash 6 or newer)
[Background color]	Background color, hexadecimal (e.g. #FFFFFF for white)
[URL and parameter]	The address to the FSI Viewer and parameters provided as HTTP query, e.g. "`?cfg=foo`" For information about available parameters please see chapter "FSI Parameters".

For further possible parameters or XHTML configuration options, please refer to Flash documentations or renowned HTML references.

NeptuneLabs FSI Viewer Configuration

Configuration parameters can be provided using **XML**- configuration files and/or by **HTTP queries** appended to the URL of the Flash movie clip.
Providing parameters by HTTP query is recommended if you plan to display large numbers of images (→Automated Implementation of Images).

In the simplest case you only define the relative path to a configuration file within the "**src**" parameter of the `<object>`-Tag using the "**cfg**" parameter:

Example – Defining the CFG parameter inside the object tag

```
<PARAM NAME="movie" VALUE="fsi.swf?cfg=[relative path to FSI-file]">
```

In general FSI parameters may contain the following value types:

Type	Example
Number	"90"
String	"ZoomIn"
URL	"http://www.fsi-viewer.com/"
Boolean	either "0" / "1" or "true" / "false"
HexColor	"FF0000"

Using XML Configuration Files (*.fsi)

Configuration files have to be created in XML- format and can easily be edited with any text editor.

The default configuration file "**_default.fsi**" - which is located in the same folder as the main viewer file "fsi.swf" – is the most important XML configuration file and will always be evaluated first.

To use FSI Viewer with additional, image specific configuration files you have to provide the name of the configuration file using the "**cfg**" parameter attached to the URL of the "fsi.swf" file:

Example – Specifying an XML configuration file for FSI Viewer

```
<PARAM name="Movie" VALUE="/fsi/fsi.swf?cfg=foo">
```

Configuration files are structured into configuration groups (XML nodes) containing parameter names and values. Only parameters in these groups will be recognized by FSI Viewer, but configuration files do not need to contain all groups.

Each parameter has to be provided as an individual XML child node of a group where the node name is the parameter name and the parameter value is the attribute "value" of the node.

Structure of XML configuration files

```
<fsi_parameter>

   <Viewer>
      <parameter value="" />
   </Viewer>

   <FPX>
      <parameter value="" />
   </FPX>

   <Plugins>
      <parameter value="" />
   </Plugins>

   <Options>
      <parameter value="" />
   </Options>

</fsi_parameter>
```

Example – Simple XML configuration file, e.g. named "foo.fsi"

```
<fsi_parameter>
   <FPX>
      <Src value="image.fpx" />
   </FPX>
   <Options>
      <Debug value="1" />
   </Options>
</fsi_parameter>
```

Please keep in mind the following rules when using XML configuration files:

- Values must be written in quotation marks
- Configuration files might be cached by your browser, so that changes in these files require flushing the browser cache. This does not apply when using the debug mode.

You can comment on or disable sections of XML configuration files using the following comment syntax:

Example - comments

```
<FPX>
   <!-- This is a comment -->
   <Src value="image.fpx" />

   <!-- The following section will be ignored -->
   <!--
    <Src value="image.fpx" />
    -->
</FPX>
```

Using HTTP Queries

In addition to using XML configuration files you can define FSI parameters by adding a HTTP query to the URL of the FSI Viewer movie clip.
Parameters have to be provided in the following form:

Parameter1=Value1&Parameter2=Value2...

Characters that are not URL compatible such as "/", "%" and "&" within the values have to be URL-encoded.

Example – Providing parameters by HTTP query

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfl
ash.cab#version=5,0,42,0" width="320" height="240">
<PARAM NAME="movie" VALUE="/fsi/fsi.swf?cfg=foo&debug=1&MenuAlign=TL">
</OBJECT>
```

As you cannot define parameter groups when providing FSI parameters by query you have to prefix parameters with the name of the corresponding parameter group.

Example – Prefixing parameters with group names

```
XML Syntax:
<FPX>
   <Src value="foo.tif" />
   <ServerType value="eRez" />
<FPX>

Equivalent HTTP Query:
?FPXSrc=foo.tif&FPXServerType=eRez
```

As most parameters belong to the "**options**" group these parameters do not need to be prefix to keep queries as short as possible.

Example – Providing <Options> parameters by query

XML Syntax:

```
<Options>
   <Debug value="true" />
   <MenuAlign value="TL" />
<Options>
```

HTTP Query equivalent:

```
?debug=true&MenuAlign=TL
```

The FSI parameter reference contains the parameter group a parameter belongs to as well as the long parameter names to be used with HTTP queries.

Hierarchy of Configuration Parameters

As explained before parameters can be provided in three different ways:

1. The default configuration file "**_default.fsi**" - located in the same folder as the main viewer file "fsi.swf" - will always be evaluated
2. You can store image specific parameters in separate *.fsi files
3. You can provide parameters by HTTP query

Parameters specified multiple times – for example in the default configuration and by query – will be evaluated in the following way:

- Parameters defined in configuration files supersede parameters defined in the default configuration file "_default.fsi".
- Parameters provided by query supersede parameters in configuration files.

Configuration Hierachy

I. HTTP Query
II. Configuration file (*.fsi)
III. Default configuration file (_default.fsi)

Choosing a Configuration Method

FSI Viewer can be configured by any combination of

- Default configuration file (_default.fsi)
- Configuration files (*.fsi)
- HTTP query

The following considerations might be helpful when deciding where to place a certain parameter:

Default Configuration
As the default configuration gets cached by the browser and you can easily change global parameters for all images by editing a single file it is the recommended configuration method for:

- Global parameters valid for all images, in particular:
  - FPXServerType
  - FPXBase
  - Licence Key
- General configuration options like
  - Plug-ins
  - MenuAlign and alike

Configuration Files
Separate configuration files can be used for parameters specific to individual images or to a group of images (e.g. "flowers.fsi", "cars.fsi" or "SmallView.fsi", "LargeView.fsi", etc.)

HTTP Query
Recommended for parameters from databases or parameters defined by user interaction. Using server side script or JavaScript you can build queries to pass parameters to FSI Viewer.

If you want to display an image with minimal variation in configuration, it is recommended to use the same FSI configuration file and to provide varying parameters by HTTP query.

Example – Providing parameters

```
View 1:
<PARAM NAME="movie" VALUE="fsi.swf?cfg=flower">

View 2:
<PARAM NAME="movie" VALUE="fsi.swf?cfg=flower&NoNav=1">
```

Retrieving Parameters Automatically

Depending on the type of imaging server being accessed, FSI Viewer is able to retrieve some image specific parameters automatically. The following table shows the parameters that can be retrieved automatically:

	YaWah eRez Imaging Server	Iseemedia Imaging Server	True Spectra Imaging Server
FPXWidth	Yes	Yes	Yes
FPXHeight	Yes	Yes	Yes
FPXTilesX	Yes (IPTC data)	---	---
FPXTilesY	Yes (IPTC data)	---	---
SceneSets	Yes (IPTC data)	---	---

When will these values be automatically retrieved?
FSI Viewer tries to retrieve the parameters FPXWidth and FPXHeight from the server if the values have not been defined manually. If the server delivers additional parameters (e.g. FPXTilesX, FPXTilesY) these parameters will only get effective if they have not been defined manually.

Requirements for Automatic Parameter Supply

1. Cross-Domain Access
As Flash movie clips can generally not access data across domain boundaries, FSI Viewer has to be setup to the same domain / IP address as the imaging server.

For imaging servers that integrate into common web browsers (eRez, TrueSpectra Bridge) this can easily be accomplished by setting up FSI Viewer to a subdirectory of the web server.

For imaging servers delivering images through their own server engine the following solutions apply:

- Setting up FSI Viewer to a web server on the same IP as the imaging server, using a different port (e.g. port 8080)
- Using NeptuneLabs ZoomCache to route FSI Viewer requests
- Setting up FSI Viewer to a subdirectory of the imaging server that is able to deliver standard file types (e.g. 'servercode' of the Iseemedia Imaging server)

2. Preventing Security Popup with Flash MX 2004™ plug-ins

From Flash MX 2004™ Macromedia modified the security model of Flash plug-ins regarding the cross domain policy. To avoid security messages popping up when FSI Viewer tries to retrieve data from the Imaging server you have to add an XML file to the **root directory** of the imaging server being requested. The file contains information about domains that are allowed to retrieve data from the specific domain and has to be named **'crossdomain.xml'**.

Example – crossdomain.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <cross-domain-policy>
  <allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

Please refer to the Macromedia® documentation for more information regarding 'cross domain security'.

Basic Parameters

The following parameters are essential and usually need to be specified.
Please refer to the chapter →"**Retrieving Parameters Automatically**" for information on how to enable FSI Viewer to retrieve image specific parameters values automatically.

CFG	
Description:	Relative path to a configuration file
Syntax:	fsi.swf?cfg=[FSI_Name]
Default:	---
Context:	HTTP query only

A relative path to an *.fsi-configurations file.
The file extension ".fsi" has to be omitted.
You can specify a default base path to all configuration files using the →FSIBase parameter.

Example - Defining a configuration file

```
<PARAM NAME="movie" VALUE="fsi.swf?cfg=foobar">
```

In this case the FSI Viewer first searches for the _default.fsi file in the installation folder where the file fsi.swf is located and evaluates the configuration parameters.

The optional parameter "**FSIBase**" from the _default.fsi file is then used as the path specification in order to search in the correct index for a FSI configuration file with the name "foobar.fsi".

DefaultCFG	
Description:	Relative path to an alternate default configuration file
Syntax:	fsi.swf?cfg=[filename]
Default:	_default
Context:	HTTP query only

Relative path to an alternate default configuration file other than "_default.fsi".
Only useful as part of an HTTP query.
The file extension ".fsi" has to be omitted.

ServerType	
Description:	Image Server type
Syntax:	String – eRez | ZIS | TrueSpectra | TrueSpectra Bridge
Default:	---
Context:	<FPX>
Query:	FPXServerType

Please enter the value "**erez**" for YaWah eRez Imaging Server.
For the Zoom Image Server of iSeeMedia please enter the value "**ZIS**".

For the Image Server of TrueSpectra please enter the value "TrueSpectra" if you are using the stand alone server and "TrueSpectra Bridge" if you are using the TrueSpectra Bridge API.

ServerTemplate	
Beschreibung:	Real-time template of the imaging server
Syntax:	String – template name
Default:	fsi
Kontext:	<FPX>
Query:	FPXServerTemplate
Version:	2.1.0 or higher

This parameter applies to eRez Imaging Servers only and specifies the name of a real-time template used when accessing the imaging server.
Using these templates you can specify e.g. the image compression, maximum image dimensions and watermarks.

Src	
Description:	URL of the source image to be displayed
Syntax:	URL
Default:	---
Context:	<FPX>
Query:	FPXSrc

Specify either a relative URL or the absolute URL of the source image on the imaging server. The parameter →**FPXBase** can be used to specify a base path to the imaging server.

Please refer to the documentation of your imaging server for details on supported image formats.

Example 1 - Absolute addressing (eRez Imaging Servers)

```
<FPX>
      <Src value="http://www.domain.com/erez3/erez?src=project/foo.tif"
/>
</FPX>
```

Example 2 - Absolute addressing (Iseemedia Imaging Servers)

```
<FPX>
      <Src value="http://www.domain.com/fif=project/foo.fpx" />
</FPX>
```

Example 3 - Partial (relative) addressing

```
<FPX>
      <Src value="project/foobar.fpx" />
</FPX>
```

Width and Height	
Description:	Source image dimensions
Syntax:	Number in pixels
Default:	---
Context:	<FPX>
Query:	FPXWidth and FPXHeight

Required parameters specifying the width and height of the source image. Please refer to the chapter "**Retrieving Parameters Automatically**" for information on how to automatically retrieve these values from the imaging server.

Example - Width and height specifications for a source image

```
<FPX>
      ...
      <Width  value="3330" />
      <Height value="4660" />
      ...
</FPX>
```

Optional Parameters

The following parameters do not need to be defined. They can be used to alter the appearance of FSI Viewer user interface.

Skin	
Description:	Filename of FSI Skin to use (without extension)
Syntax:	String
Default:	---
Context:	<Options>

Filename of FSI Skin file to use (omit the file extension). FSI Skins determine the appearance of the user interface and are located in the subdirectory "/skins".
In addition to the default skins that ship with FSI Viewer customized skins are available on demand.
Please note that using unregistered skins activates the evaluation mode.

Intro	
Description:	Filename of start animation to use (without extension)
Syntax:	String
Default:	---
Context:	<Options>

Filename of animation to display on startup (omit the file extension).
FSI Viewer ships with a default start animation. Custom intros are available on demand.
Please note that using an unregistered intro activates the evaluation mode.

Debug	
Description:	Activate debug mode
Syntax:	Bool
Default:	false
Context:	<Options>

Activate the debug mode providing configuration details, warnings and error messages. It is strongly recommended to activate the debug mode when implementing new images with FSI Viewer.
Please refer to the chapter →**Debug Mode** for a detailed description.

FSIBase	
Description:	Path to FSI configuration files
Syntax:	Relative path
Default:	---
Context:	<Options>, _default.fsi only

Enter the relative path from FSI Viewer to the image specific configuration files here. If this parameter is specified the path is added to all **relative** CFG parameters.

Example 1 - Definition of FSIBase in the _default.fsi file

```
<Options>
      ...
      <FSIBase value="config/" />
      ...
</Options>
```

Example 2 - Specifying FSI files using the CFG parameter

```
<OBJECT ...>
    ...
    <PARAM NAME="movie" VALUE="/fsi/fsi.swf?CFG=imagedir/image1">
    ...
</OBJECT>
```

The content of the parameter "**CFG**" will be extended by the parameter "**FSIBase**" to "`config/imagedir/image1.fsi`"

Base	
Description:	Image Server URL
Syntax:	URL
Default:	---
Context:	<FPX>

Domain and path to the imaging server.
All "**FPXSrc**" parameters containing relative paths (without "http://...") will be extended by the path defined by "**FPXbase**".

Examples - FPXBase application

Definition of FPXBase within the _default.fsi file:

```
<FPX>
      ...
      <Base value="http://domain.com/erez?src= " />
      ...
</FPX>
```

```
Specification of the FPXSrc within configuration files:

1) FPXBase parameter is used for relative addressing:
<FPX>
      ...
      <SRC value="folder/image.tif" />
      ...
</FPX>

The resulting address is:
http://domain.com/erez?src=folder/image.tif

2) FPXBase is not used for absolute addressing:
<FPX>
      ...
      <SRC value="http://domain.com/erez?src=project/image.tif" />
      ...
</FPX>
```

Width and Height	
Description:	Size of the Flash movie clip
Syntax:	Number in pixels
Default:	---
Context:	<Viewer>
Query:	ViewerWidth and ViewerHeight

Width and height of the viewer in pixel.
These parameters are **only necessary for Flash™ 5 plug-ins** and must match the width and height parameters in the <object> or <embed> tag.
Providing incorrect values leads to skewed or incorrectly scaled presentations of FSI Viewer as far as Flash™ plug-ins version 5 are concerned.

For Flash MX or higher you can either leave out the parameters or enter "auto".

Border	
Description:	Size of the Flash movie clip
Syntax:	HexColor or false
Default:	---
Context:	<Viewer>
Query:	ViewerWidth and ViewerHeight

Display a thin border around FSI Viewer. The default value is "False" displaying no border. Enter one 6-digit hexadecimal color (e.g. "FF0000") for a solid border. If you specify two hexadecimal colors (e.g. "FF0000, 00FF00") the top left border will use the first color while the bottom left border uses the second color.

MenuAlign	
Description:	Alignment of the menu bar
Syntax:	String
Default:	TL
Context:	<Options>

Possible values:

TL (top-left)
TR (top-right)
BL (bottom-left)
BR (bottom-right)

Language	
Description:	Interface language
Syntax:	String
Default:	english
Context:	<Options>
Version:	FSI Viewer 2.0 or higher / FSI Showcase 2.15 or higher

Defines the language of tool tips for the user interface.
Possible values depend on the language files located in the sub-index "/languages" of the FSI Viewer setup path. To specify a language, please enter the filename without file extension, e.g. "german".

Some language files contain the pattern "**_font**" in the filename. These files include a corresponding font for the specified language. Though the size of these files is greater than those without an embedded font, the included font ensures correct display on systems with character sets not matching the selected language.

DetailBuffer	
Description:	Enable or disable the detail buffer
Syntax:	Boolean
Default:	true
Context:	<Options>
Version:	3.0.0 and above

Enabling the detail buffer keeps previously loaded image details visible when zooming in or out instead of displaying the low resolution preview image while loading additional image details. This way zoom transitions appear smoother.

NoNav	
Description:	Hide the survey window
Syntax:	Bool
Default:	false
Context:	<Options>

Show or hide the survey window displaying the miniature image and the current image section.

NavWidth and NavHeight	
Description:	Maximum size of the survey window
Syntax:	Number in percent or pixel
Default:	80
Context:	<Options>
Version:	3.0.0 and above

Defines the maximum width and height of the image inside the survey window. The value can be defined absolutely or in percent of the viewer size.
Use parameter →**NoNav** to hide the survey window.

NavFrame	
Description:	Show the selection frame of the survey window
Syntax:	Boolean
Default:	true
Context:	<Options>
Version:	3.0.0 and above

Specifies whether to show (true) or hide (false) the selection frame of the survey window.

NavFrameColor	
Description:	Color of the selection frame inside the survey window
Syntax:	HexColor
Default:	FF0000 (depending on the skin)
Context:	<Options>
Version:	3.0.0 and above

Specify a 6-digit HexColor to change the color of the selection frame inside the survey window, e.g. "0000FF" for blue.

NavMaskColor	
Description:	Color of the area around the selection frame of the survey window
Syntax:	HexColor
Default:	depends on the skin
Context:	<Options>
Version:	3.0.0 and above

Specify a 6-digit hexadecimal color to change the color of the area outside the selection frame of the survey window, e.g. "FFFFFF" for white.

NavMaskAlpha	
Description:	Opacity of the area around the selection frame of the survey window
Syntax:	Number (0..100)
Default:	depends on the skin
Context:	<Options>
Version:	3.0.0 and above

Specify the opacity of the area outside the selection frame of the survey window. Possible values are integer values from "0" (invisible) to "100" (opaque).

InitialView	
Description:	Initial image segment
Syntax:	SceneSet, Scene [left, top ,right, bottom, rotation]
Default:	1,1,0,0,1,1,0
Context:	<Options>

Image section (and rotation) to display on startup.
The first two parameters are required as they specify the scene set and the Scene. For 2D image both of these values default to 1.
The other parameters are optional and specify the image segment and the rotation around the z-axis. Please use the plug-in →**SelectFrame** to easily retrieve valid values for this parameter.

Example - InitalView

```
<Options>
      ...
      <InitialView value="1,5,0,0,0.25,0.25,90" />
      ...
</Options>
```

The example displays the above left quarter of the 5th scene in the 1st scene set rotated 90° to the right.

InitialViewPersistent	
Description:	Keep the InitialView as default view
Syntax:	Bool
Default:	false
Context:	<Options>
Version:	2.0.0 or higher

If this parameter is activated, clicking the "Reset" button will display the image section defined by the InitialView parameter instead of the entire image.

If this parameter is set to "false", the initial view will be displayed on startup, but pressing the Reset button will reset the view to the entire image.

InitialAction	
Description:	Action on startup
Syntax:	String
Default:	---
Context:	<Options>

Specifies an action to execute on startup. The action is repeated until the user presses a button or key.
While the action is being executed no image details are loaded.

Possible values:

NextScene (rotate around the y-axis to the right, 3D only)
PreviousScene (rotate around the y-axis to the left, 3D only)
NextSceneSet (rotate around the x-axis to the right, 3D only)
PreviousSceneSet (rotate around the x-axis to the left, 3D only)
RotateRight (rotate around the z-axis to the right, 3D only)
RotateLeft (rotate around the z-axis to the left, 3D only)
ZoomIn (zoom in)
ZoomOut (zoom out)

InitialActionDelay	
Description:	Speed of the initial action
Syntax:	Number
Default:	3
Context:	<Options>

Specifies the speed of the action specified by the InitialAction parameter.
Possible values are integer values greater than 0.

0 highest speed
>1 slower

InitialActionPersistent	
Description:	Repetition of initial action
Syntax:	Bool
Default:	false
Context:	<Options>
Version:	2.2.0 or higher

Initial action restarts on each 'reset' command (button or keyboard) if set to true. Otherwise the action only performs on startup.

HelpURL	
Description:	URL of custom help page
Syntax:	String
Default:	http://help.fsi-viewer.com
Context:	<Options>
Version:	1.3.0 or higher

If you want to create a custom help page for your FSI Viewer enter the complete (absolute) URL to this page using this parameters.

HelpURLTarget	
Description:	Target (frame) for custom help page
Syntax:	String
Default:	_blank
Context:	<Options>
Version:	1.3.0 or higher

Using this parameter you can define the target name of the browser window if a user opens the help page. The default value "_blank" opens a new browser window.

InitialMouseMode	
Description:	Mouse mode on startup
Syntax:	Number
Default:	0
Context:	<Options>

Specifies the selected mouse mode on startup.

Possible values:
0 (Zoom)
1 (Pan)

2 (Rotate 3D)
3 (Rotate 2D)

Some plug-ins provides additional mouse modes. Please refer to the corresponding plug-in reference for details.

HandCursor	
Description:	Shape of the mouse cursor
Syntax:	Bool
Default:	true
Context:	<Options>
Flash Version	No effect with Flash versions prior to version 6

If you set this parameter to "true" the default arrow cursor will be used instead of the hand cursor when hovering over the image or the survey window.

ZoomLimit	
Description:	Maximum magnification
Syntax:	Number
Default:	100
Context:	<Options>

Defines the maximum magnification in percent relative to the source image.
The default value "100" permits magnification to the physical resolution of the source image. Possible values are "1" to "1000".

ZoomLimitMin	
Description:	Minimum magnification
Syntax:	Number
Default:	100
Context:	<Options>

Defines the minimum magnification in percent relative to the initial image.
The default value "100" permits magnification to the size of the Viewer. Smaller values permit magnifications smaller than the viewer size.
Possible values are "1" to "100".

ExtendedViewport	
Description:	Extend the size of the view port
Syntax:	Boolean
Default:	False
Context:	<Options>

With this option enabled you can pan the borders of the image to the center of the view port. If you disable this option the panning range is smaller so that the user can not pan the image borders to the center of the view port.

Animation	
Description:	Animation quality
Syntax:	String
Default:	best
Context:	<Options>

Specifies the quality during animations (transitions).
The lower the quality, the smoother the animations.

Possible values:

none (no animation at all)
low (low quality, high speed)
medium (good quality and speed)
high (very good quality, slower speed with slow computers)
best (best quality, slower speed with slow computers)

AnimationSpeed	
Description:	Animation speed
Syntax:	Number
Default:	50
Context:	<Options>

Specifies the speed of the zoom and scene animations.

Possible values range from 1 (very slow) to 100 (very fast).
See parameter "**Animation**" to disable animated motion.

HideUI	
Description:	Hide user interface
Syntax:	Bool
Default:	false
Context:	<Options>

Hide the menu bar on startup. The user has to click the "Show Menu" button to show the user interface. This option is especially useful when an image is meant to look like an ordinary image, or when there are many FSI Viewers on a single web page.

UISwitchable	
Description:	Enable hiding/showing the user interface
Syntax:	Bool
Default:	True
Context:	<Options>
Version:	3.2.2 and above

Specifies whether the user is able to show or hide the user interface by clicking the corresponding button or pressing "0" of the num pad. The "Show Menu" button is hidden if this option is set to "false".

NoImageBlend	
Description:	Subsequently loaded image details display without transition
Syntax:	Bool
Default:	False
Context:	<Options>
Version:	1.2.0 or higher

If this parameter is activated, subsequently loaded image details are displayed abruptly without transition.

Effects	
Description:	Image manipulation parameters for all images
Syntax:	String
Default:	---
Context:	<Options>
Version:	1.3.0 or higher

The effects parameter can be used to pass image manipulation parameters to the imaging server. The range of available parameters depends on the imaging

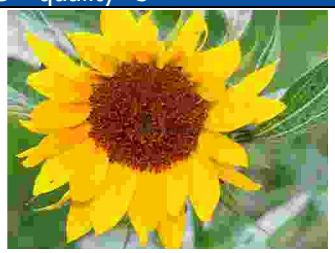
server being used. Typical manipulation parameters include the image compression ("`quality`" or "`qlt`") and image sharpening/blurring ("`sharpen`" or "`ftr`"). If no effect is specified, the default settings of the server apply. Please refer to your imaging server for a list of available parameters.

Example "quality"

The "quality" or "qlt" parameter specifies the JPEG compression of images delivered by the imaging server.

Practical values range from 50 to max. 90

Example – quality=5



```
<effects value="quality=5" />
```

Example – quality=90

```
<effects value="quality=90" />
```

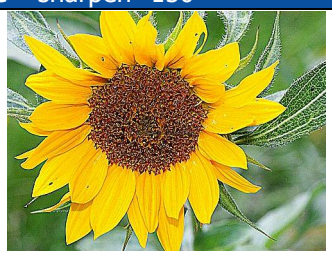
Example "sharpen"

The "sharpen" or "ftr" parameter blurs (negative values) or sharpens (positive values) images delivered by the imaging server.

Example – sharpen=-300 (blur)

```
<effects value="sharpen=-300" />
```

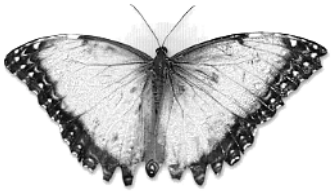
Example – sharpen=150



```
<effects value="sharpen=150" />
```

Other effects change, for example, the color saturation, hue or contrast. Please refer to your imaging server documentation for a complete description of available image effect commands.

Example – saturation=0



```
<effects value="saturation=0" />
```

Example – hue=180&cnt=1.4

```
<effects value="hue=180&cnt=1.4"/>
```

The format of the effects parameter is similar to a HTTP query. You can combine different effects by concatenating the "parameter=value" pairs with an ampersand (&) character.

```
<effects value="quality=65&sharpen=30"/>
```

When passing effects by HTTP query the value of the effects parameter has to be provided as an url-escaped string (e.g. "%26" instead of "&" and "%3D" instead of "=").

```
...fsi.swf?effects=quality%3D65%26sharpen%3D30
```

In general FSI Viewer passes the value of the effects parameter through to the imaging server. This is you can pass through all parameters supported by your imaging server valid for static images using the effects parameter.

For example with eRez imaging server you can force the use of a real-time template by passing the "`tmp=`" parameter:

```
<effects value="tmp=fsi2"/>
```

Applying Effects to Specific Images Only

While the "effects" parameter holds image manipulation parameters for all images you can as well apply effects to specific types of images in FSI Viewer (see "NavEffects", "SceneEffects" and "TileEffects").

For example it is sometimes useful to sharpen the images inside the survey window as small images tend to appear a little blurry. To do so you can add a corresponding "NavEffects" parameter.

Sharpening Images in the survey window:

`<NavEffects value="sharpen=35"/>` (eRez Imaging Server)
`<NavEffects value="ftr=5"/>` (Iseemedia Zoom Image Server)

NavEffects	
Description:	Image manipulation parameters applied to the survey window
Syntax:	String
Default:	---
Context:	<Options>
Version:	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all images inside the survey window. Please refer to "effects" parameter for a detailed description.

SceneEffects	
Description:	Image manipulation parameters applied to preview images
Syntax:	String
Default:	---
Context:	<Options>
Version:	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all un-tiled images in the viewer. Please refer to "effects" parameter for a detailed description.

TileEffects	
Description:	Image manipulation parameters applied to image tiles
Syntax:	String
Default:	---
Context:	<Options>
Version:	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all images tiles in the main view area. Please refer to "effects" parameter for a detailed description.

ZoomCache	
Description:	URL of the ZoomCache
Syntax:	URL
Default:	---
Context:	<Options>

If you use NeptuneLabs ZoomCache, please specify the URL of the ZoomCache server here. If you haven't setup one yourself, you can receive this URL and a ZoomCache ID from your image hosting provider.

ZoomCacheID	
Description:	ID of the ZoomCache access profile
Syntax:	String
Default:	---
Context:	<Options>

When using NeptuneLabs ZoomCache enter the name of the access profile for your images here. You may have several profiles i.e. if you use watermarking or different imaging servers.

ZoomCache5Only	
Description:	Only use ZoomCache with Flash 5 plug-ins
Syntax:	Bool
Default:	false
Context:	<Options>

Use ZoomCache only for Flash-Plug-ins version 5.

IgnoreQuery	
Description:	Ignore all parameters passed via query (except **cfg**)
Syntax:	Bool
Default:	false
Context:	<Options>
Version:	2.0.0 or higher

If this parameter is activated, the viewer ignores all parameters passed via HTTP-query except for the "**cfg"** parameter.
This way you can disable modification of FSI configuration by query.

MaxZoomLevel	
Description:	Load image details at maximum zoom
Syntax:	Boolean
Default:	True
Context:	<Options>
Version:	3.2.1 or higher

By default FSI Viewer adds a zoom level for the maximum magnification – this is FSI Viewer loads image details when reaching the maximum magnification. For extremely high resolution 2D images exceeding 32,000 pixel in width or height you can set this value to "false" to prevent display problems at the maximum magnification.

TileSizeX and TileSizeY	
Description:	Size of image detail tiles
Syntax:	Number in pixel or in percent of the viewer size
Default:	50%
Context:	<Options>
Version:	2.3.0 or higher

Specifies the size of image details (image tiles). The value can be defined relative to the viewer size (e.g. "25%") or using absolute pixel values (e.g. "256").
The default value of 50% means that each image tile is one quarter in size of the entire viewer. This way a maximum of 9 image tiles is required to display the selected image section.

Please do not change this value unless you are certain that this is necessary and you are aware of possible consequences:

- Defining smaller image tiles reduces the overall viewer performance and increases the number of requests and thus the imaging server load.
- Defining larger image tiles might lead to increased traffic and decreased download performance as more image data than actually required for a certain image section will be loaded.

ConcurrentRequests	
Description:	Number of simultaneous image requests
Syntax:	Number
Default:	2
Context:	<Options>
Version:	2.3.0 or higher

Specifies the number of image details (image tiles) being requested simultaneously.

Please do not change this value unless you are certain that this is necessary and you are aware of possible consequences:

- Increasing this value might lead to increased imaging server load
- Decreasing the value will decrease the download performance

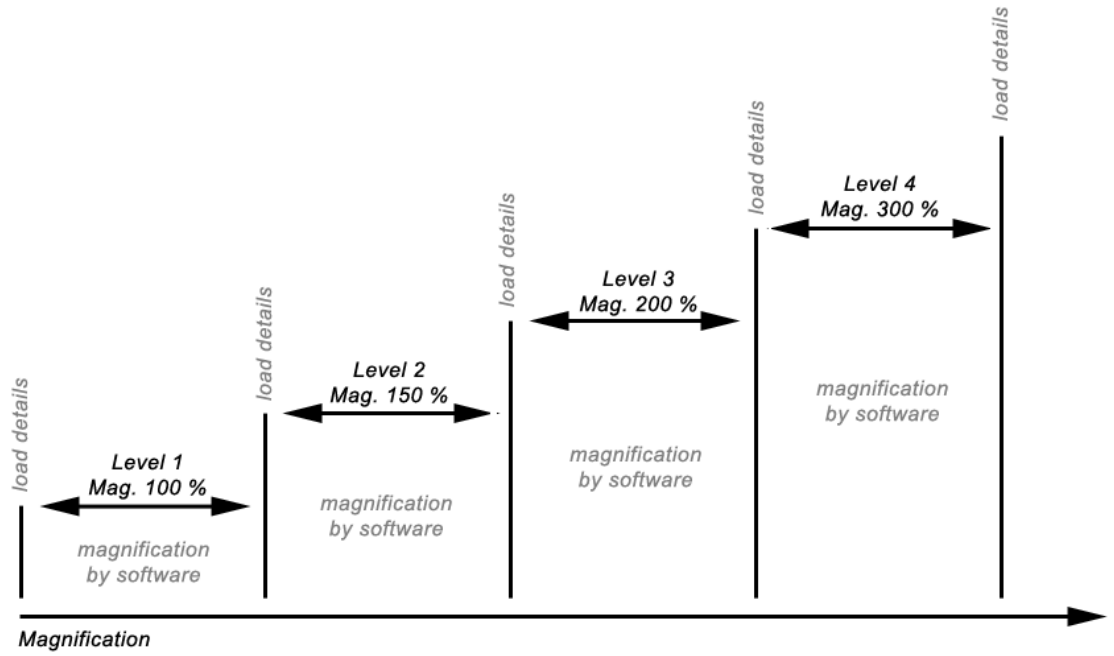
ZoomLevelFrequency	
Description:	Number of Zoom Levels
Syntax:	Number
Default:	2
Context:	<Options>
Version:	2.3.0 or higher

To enable effective caching of image data FSI Viewer does not load image data for every level of magnification. Instead FSI Viewer uses dynamic zoom levels to determine when to load image data and when to scale the image by software. By modifying this value you can define the frequency of zoom levels.

Please do not change this value unless you are certain that this is necessary and you are aware of possible consequences:

- Defining higher values will instruct FSI Viewer to load image data more frequently. This decreases the number of cache hits and increases data traffic, On the other hand higher values increase image quality as software image scaling will be reduced.
- Defining smaller values increases cache hits and decreases the data traffic volume. Lower values might lead to reduced image quality though.

Possible Values are integer or floating point number greater or equal to 1.



The figure above displays a zoom level frequency of 2.

NeptuneLabs FSI Viewer
Virtual 3D Presentations

In addition to zoomable 2D images FSI Viewer can as well be used to display virtual 3D presentations of an object. In this case the source image must contain multiple images (tiles) displaying the same object after different steps of rotation. The number of horizontal and vertical image tiles must be specified using the parameters →**TilesX** and **TilesY**.

Figure A displays a source image containing 12 tiles representing a rotation of 360° around the Y-axis in steps of 30°. In this case you simple add the parameters `TilesX=3` and `TilesY=4` to your FSI configuration.

Fig A: Rotation around Y-axis in 30° steps
3 x 4 = 12 tiles

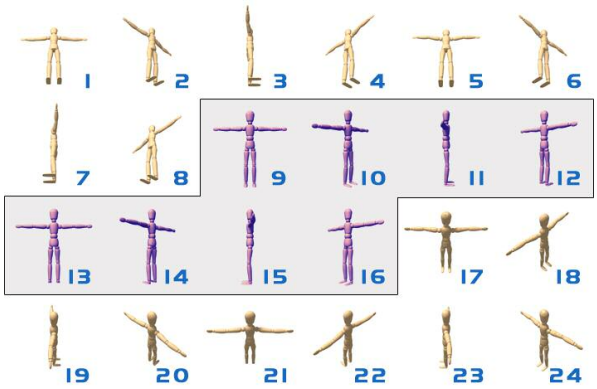


Fig B: Rotation around Y-axis in 45° steps and partial rotation around X-axis in 15° steps
6 x 4 = 24 tiles in 3 scene sets

Example – Configuration for a 3D presentation as displayed in **Figure A**

```
<FPX>
   ...
   <TilesX value="3" />
   <TilesY value="4" />
</FPX>
```

Figure B shows a more complex 3D view representing a rotation of 360° around the X-axis in 45° steps and an additional rotation of +/- 15° around the Y-axis.

To present rotations around more than one axis or to present an object in different states you need to define →**Scene Sets**.

A Scene Set describes a sequence of individual images.
The user can select individual images of a scene set using the keys "7" or "9" in the num block or using the mouse mode "rotate" moving the mouse horizontally. The user can additionally select a scene set using the keys "/" and "*" in the num block or using the mouse mode rotate moving the mouse vertically.

If the **TilesX** or **TilesY** parameters are defined for a source image, but no scene set is defined, a scene set containing all the scenes from left to right, top to bottom will be created by default. Referring to **Figure B** this would be

```
<SceneSets value="1-24" />
```

As the presentation of **Figure B** consists of 6 by 4 individual images while 8 images represent a complete rotation around the Y-axis the definition of the following 3 scene sets is required:

Image Tiles	Y-Axis	X-Axis
1 – 8	0 to 360°	- 15°
9 – 16	0 to 360°	0°
17 – 24	0 to 360°	+15 °

The required configuration to rotate the object around the Y-axis moving the mouse horizontally and rotating the object around the X-axis moving the mouse vertically looks like this:

Example – Configuration for a 3D view as displayed in **Figure B**

```
<FPX>
   ...
   <TilesX value="6" />
   <TilesY value="4" />
</FPX>

<Options>
   ...
   <SceneSets value="1-8,9-16,17-24" />
   ...
</Options>
```

Please refer to the description of the **SceneSets** parameter for details on the different ways of defining images contained in a scene set.

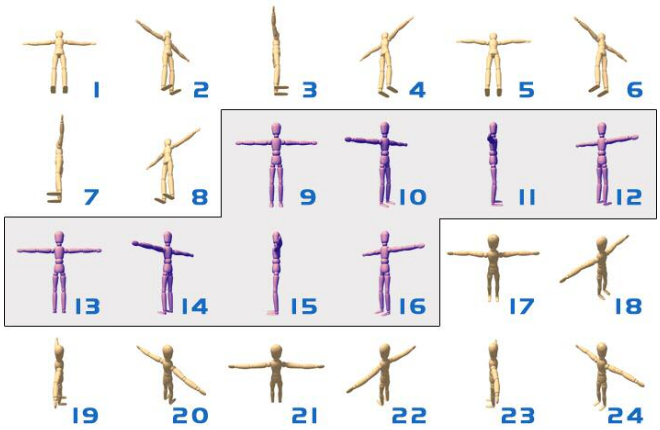
Tip: If you are not sure about the number of images in your source image in the x and y direction, firstly enter the value 1 for **TilesX** and **TilesY**. In the viewer you will then see the entire source image structure.

3D Parameters

The following FSI configuration parameters apply to virtual 3D presentations only.

TilesX and TilesY	
Description:	Number of individual images for 3D presentations
Syntax:	Number
Default:	1
Context:	<FPX>
Query:	FPXTilesX and FPXTilesY

Number of individual images contained in the source image in x- and y- direction respectively.



Example: 3D image containing 6 x 4 tiles arranged in 3 scene sets.

Even with 'asymmetrical' scene orders - like in the example to the left -the number of tiles would be 6 in the x direction and 4 in the y direction.

Example - Definition of 3 scene sets containing 6 x 4 = 24 tiles

```
<FPX>
      ...
      <TilesX value="6" />
      <TilesY value="4" />
      ...
</FPX>
<Options>
      ...
      <SceneSets value="1-8,9-16,17-24" />
      ...
</Options>
```

SceneSets	
Description:	Scene sequence definition (3D)
Syntax:	String
Default:	1 – n tiles
Context:	<Options>

Using this parameter you can define scene sets for images containing multiple scenes (→TilesX, TilesY). Scene sets define the order of scenes to be displayed. The default is a single scene set containing all scenes (image tiles) from scene 1 to the last scene image (left to right, top to bottom).
In the viewer scenes can be selected by moving the mouse horizontally and scene sets can be selected by moving the mouse vertically.

Syntax:
Individual sets have to be separated with ";" (semicolon).
Scenes within the sets are separated with "," (comma) and specify the 1-based index of the image from left to right, top to bottom up to the total number of scenes (TilesX x TilesY).

Instead of individual scenes you can also specify scene ranges, e.g. "5-10".
If you leave out the start or the finish of a scene range, 1 or respectively the last scene will be assumed. If you specify a range using "->" the scenes from the first number to the last scene image plus the first scene image up to the second number will be assumed.

Example - Definition of multiple scenes and scenesets

```
<Options>
      ...
      <SceneSets value="2,3,5-10;10-1;5-;8->3" />
      ...
</Options>
```

The example defines 4 scene sets in total. Assumption: The source image is created from 10 individual images (scenes), 5 in a row, two rows.

This leads to the following image order for the individual scenes:

Set 1 - Scenes: 2,3,5,6,7,8,9,10
Set 2 - Scenes: 10,9,8,7,6,5,4,3,2,1
Set 3 - Scenes: 5,6,7,8,9,10
Set 4 - Scenes: 8,9,10,1,2,3

You can review the resulting scene sets in the "Info" section of the Debug Window.

ScenePreload	
Description:	Load scenes in high resolution (3D)
Syntax:	Bool
Default:	false
Context:	<Options>

If this parameter is activated all scenes will be loaded on startup in high resolution. Otherwise preview images will be loaded with lower resolution. This parameter can significantly influence loading times and traffic volumes.
Setting this parameter to false leads to faster startup times and scene images will be loaded as soon as they need to be displayed.

TilePaddingX and TilePaddingY	
Description:	Crops scene images by a given number of pixel
Syntax:	Number (pixel) or percent
Default:	0
Context:	<FPX>
Version:	2.2.0 or higher

Specifies a positive number of pixels that each source scene image (2D and 3D) will be cropped by. This parameter can be used to eliminate image margins especially with 3D objects containing polychrome backgrounds.
You can specify the padding either absolutely as a number of pixel or in percent of the source image.

NoSceneAnimation	
Description:	No SceneSet animations (3D)
Syntax:	Bool
Default:	false
Context:	<Options>

Disable animations between scenes and scene sets. This can be useful if different scenes represent different states of an object instead of a rotation.

NoSetLoop	
Description:	No 360° rotation around the y-axis (3D)
Syntax:	Bool
Default:	false
Context:	<Options>

The first and the last scene set will be considered ending points.
(E.g. for rotations < 360°)

NoSceneLoop	
Description:	No 360° rotation around the x-axis (3D)
Syntax:	Bool
Default:	false
Context:	<Options>

The first and last scene in the each scene set will be considered ending points. (E.g. for rotations < 360°)

NeptuneLabs FSI Viewer
Debug Mode

Enabling the Debug Mode

The debug mode can be enabled or disabled by passing the "Debug" parameter either by HTTP query or inside an XML configuration file.

Use "`debug=1`" to enable the debug mode and "`debug=0`" (or omit the debug parameter) to disable the debug mode.
To disable the debug mode permanently you can safely delete the "/debug" directory of your FSI setup directory.

Using the Debug Window

In debug mode an output window is available presenting information about the initialization progress, configuration parameters, plug-ins and image information.

You can select different debug sections by clicking the corresponding tab at the bottom of the window.

The window can be moved by dragging the window title bar and you can show or hide the window by pressing "d".

Enabling the debug mode additionally **prevents the browser from caching FSI configuration files**. This allows you to easily test changes to your FSI configuration files without having to delete the browser cache each time.

General section
Displays general information about the initialization progress.
If the viewer does not display an image you should be able to track the error cause by determining the step where the initialization process stops.

For example if the last output is "Loading Skin and Module" the skin file is most likely missing or corrupt.

Config section
Displays information on the configuration parsing process, the resulting configuration and possible configuration errors.

If the viewer loads successfully, but the configuration does not look as expected you should be able to track the reason in this section.
This section lists all recognized parameters resulting from the 3 different configuration methods (default configuration, configuration file and query) and displays tips regarding false parameter values as well as unknown parameters.

Plug-ins section
Displays information on loaded FSI Plug-ins and configuration errors related to plug-ins.

Info section
Displays version information, general information related to the viewer instance and information on the image being displayed.

Two pieces of information displayed here can be especially useful when implementing FSI Viewer into websites:

- **Image Section**
  Displays a string representing the current image section that can be used for the "**InitialView**" parameter and when defining hot spots for the **HotSpot plug-in**.
  Please refer to the "InitialView" parameter for an explanation of the string format.
  Press "i" to refresh the image section being displayed after zooming, panning or rotating the image.

- **Viewer Size matching Image Size**
  Displays an alternate viewer size exactly matching the image dimension. The values are based on the current width and height of the viewer.
  If you choose one of the proposed viewer sizes the image will cover the entire view port.

Debugging load errors
When enabling the debug mode FSI Viewer outputs error information if problems retrieving images from the imaging server occur.
Please note that this feature requires the imaging server to reside in the same domain as FSI Viewer or using Flash plug-in version 7 together with a valid →"crossdomain.xml" file copied to the servers root directory.

NeptuneLabs FSI Viewer
Automated Implementation of Images

Rather than creating individual configuration files for each image it is recommended to provide FSI parameters via HTTP queries appended to the FSI Viewer URL when implementing large numbers of images.
As some browsers require the obsolete <embed> tag, queries have to be added to both the <object> and the <embed> tag. (→HTML-Source Code).

FSI Viewer is able to request image specific parameters like the source image dimension (parameters "FPXWidth" and "FPXHeight") from your imaging server (→ Retrieving Parameters Automatically). Using this feature reduces the required work to dynamically providing the path to the source image (parameter "FPXSrc").

It is recommended to use server side script like ASP, PHP, JSP, Perl, etc. when implementing large numbers of images. All subsequent code examples in this section refer to PHP, but can be easily adapted in any other server side scripting language.

Steps to be taken when using large numbers of images:

1. **Defining default parameters**
   Be sure to provide as many recurring parameters as possible in the **_default.fsi** file.
   This way you don't have to pass these parameters via query.
   The most important parameter is FPXBase, as this parameter enables you to subsequently define FPXSrc parameters using relative addressing.
   If there are some images that require different parameter values than defined in the _default.fsi file, you can still overwrite the default setting by passing the value via query.

2. **Creating image specific URLs**
   Be sure to have image specific data available for server scripting by using a database, your content management system, or any other appropriate source. You will at least need to provide the FPXSrc parameter from your data source.

   After collecting the image specific data from your data source you have to build a standard HTTP query string (RFC1738) containing the data using server side scripting (→Using HTTP Queries).

Example – Generating FSI queries (fetch_fpx_image.php)

```
// Retrieve FPX properties for picture 1 from table "FPX_Images"
$result = mysql_query('select FPXSrc,FPXHeight,FPXWidth,TilesX,TilesY
from FPX_Images where PictureID=1');

if ($result) {

   // Fetch SQL data
   $fpx_parameter=mysql_fetch_assoc($result);

   if ($fpx_parameter){

       // RFC1738 parameter encoding
       foreach ($fpx_parameter as $fpx_key => $fpx_val) {
           $query_collection[]= $fpx_key.'='.rawurlencode($fpx_val);
       }

       // Make one large query string
       $fsi_query=implode('&', $query_collection);

       // Add Query to FSI Viewer URL
       $fsi_url='fsi.swf?'.$fsi_query;
   }
}
```

3. Creating the HTML-Code dynamically

Create a variable that contains the <object> and <embed> tag with all variables provided as script variables.
Replace all variables inside the template using server side scripting.

You will usually need to replace the following 4 variables:

- **URL** (→2. Creating image specific URLs)
- **Width**
- **Height**
- **Background Color**

Example – HTML Template

```
<HTML>
<BODY>
...

<?php
include("fetch_fpx_image.php");

$fsi_url  = "fsi/fsi.swf?FPXSrc=image.fpx&FPXTilesx=4&FPXTilesY=3";
$width    = 320;
$height   = 300;
$bgcolor  = "#FFFFFF";

$template="
<OBJECT classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"
codebase=\"http://download.macromedia.com/pub/shockwave/cabs/flash/swf
lash.cab#version=5,0,42,0\" width=\"$width\" height=\"$height\">
<PARAM NAME=\"movie\" VALUE=\"$fsi_url\">
<PARAM NAME=\"bgcolor\" VALUE=\"$bgcolor\">
<PARAM NAME=\"menu\" VALUE=\"false\">
<EMBED TYPE=\"application/x-shockwave-flash\"
PLUGINSPAGE=\"http://www.macromedia.com/shockwave/download/index.cgi?P
1_Prod_Version=ShockwaveFlash\" SRC=\"$fsi_url\" WIDTH=\"$width\"
HEIGHT=\"$height\" BGCOLOR=\"$bgcolor\" MENU=\"false\"></EMBED>
</OBJECT>";

echo $template;
?>

...
</BODY>
</HTML>
```

NeptuneLabs FSI Viewer
Plug-ins

FSI Plug-ins dynamically extend the functionality of FSI Viewer and can be included at runtime. Plug-ins are available from version 2.0.

Depending on the plug-in additional buttons might be added to the menu bar. The sequence of these buttons can be defined by the sequence of the plug-in definitions inside the <Plugins> section of the configuration file.

Using Plug-ins

To include a plug-in you simply list the desired plug-ins in the <Plugins> section of any *.fsi configuration file (or the _default.fsi file).
The required plug-in files are located in the "/plugins" folder of the FSI setup directory. Plug-in parameters have to be specified as attributes of the plug-in tag.

Example – Using FSI Plug-ins

```
<fsi_parameter>
...
<Plugins>
   <Plugin src="mousemodes" />
   <Plugin src="history" length="10" />
</Plugins>
...
<fsi_parameter>
```

The "**Src**" attribute which applies to all plug-ins specifies the plug-in to be included:

Src	
Description:	Name of the plug-in
Syntax:	String
Default:	---
Context:	<Plugins>
Version:	2.0.0 or higher

This attribute defines the name of the plug-in, e.g. "history".
The value has to be stated without the file extension ".plg".

Example – Using the Src Parameter

```
   <Plugin src="mousemodes" MenuOffset="10" />
```

Alternatively you can integrate plug-ins by HTTP query using the "**plugins**" parameter with a comma separated list of the desired plug-ins.

This option is available from FSI version 3.0 only.

Example – Integrating plug-ins using HTTP queries

```
<PARAM NAME="movie" VALUE="/fsi/fsi.swf?plugins=mousemodes,history">
```

The example above integrates the plug-in "mousemodes" and the plug-in "history".

Deactivating Plug-ins

To disable previously integrated plug-ins (e.g. plug-ins defined in the _default.fsi file) you can set a parameter with the name of the plug-in and a value of "false" to the <Options> section of your configuration file or provide a corresponding parameter via HTTP query.

Example – Disabling the History Plug-in

```
<Options>
...
<History value="false" />
...
</Options>
```

or by HTTP query:

Example – Using a Query to disable History Plug-in

```
<Object ...
<PARAM NAME="movie" VALUE="fsi.swf?cfg=image.fsi&history=false">
...
</Object>
```

Plug-in Parameters

Plug-in parameters are defined as attributes of the <Plugin> tag.

Please note that some plug-ins might require specific attributes or even entire XML sections in the configuration file to work properly. For information on a specific plug-in please refer to the Plug-in Reference.

Example – Defining plug-in parameters

```
<Plugins>
...
<Plugin src="navextension" MenuOffset="10" visible="true" />
...
</Plugins>
```

Providing Plug-in Parameters by Query

To provide plug-in parameters by query you have to prefix the corresponding parameter with the name of the plug-in and an underscore character:

PluginName_ParameterName=value

Example – Providing plug-in parameters via HTTP query

```
fsi.swf?navextension_visible=true&history_length=10
```

The example above sets the "visible" parameter of the "NavExtension" plug-in to "true" and the "length" parameter of the "History" plug-in to "10".

NeptuneLabs FSI Viewer
FSI Showcase Add-on

FSI Showcase Add-on integrates a resizable scroll pane to FSI Viewer displaying thumbnails of multiple images. Clicking a thumbnail displays the image in FSI Viewer.
This way a single viewer instance can be used to display hundreds of images presenting thumbnails and optional texts for each image.

Integrating FSI Showcase

The integration of FSI Showcase is very similar to an ordinary FSI plug-in. To enable the showcase you have to add a plug-in node to your (default) XML configuration file:

Example – Integrating FSI Showcase

```
...
<Plugins>
   ...
   <Plugin src="showcase" basecolor="58A8FF"/>
   ...
</Plugins>
...
```

Please note that FSI Showcase requires Macromedia Flash™ plug-in version 6 (Flash MX) or above.

Defining Image Lists

The images to be displayed in FSI Showcase have to be defined in an <images> node added to your XML configuration file.

Example – Defining FSI Showcase image nodes

```
...
<Images>
   ...
   <Image label="My First Image" file="image1.fsi" />
   <Image label="My Second Image" file="image2.fsi" />
   ...
</Images>
...
```

There are three different ways of defining image lists:

1. **Creating image nodes referencing external XML configuration files**

Example – External XML configuration files

```
...
<Images>
   ...
   <Image label="My First Image" file="image1.fsi" />
   <Image label="My Second Image" file="image2.fsi" />
   ...
</Images>
...
```

2. **Creating image nodes containing complete image configurations**

Example – Complete XML configurations

```
...
<Images>

   <Image label="My First Image">
      <FPX>
         <SRC value="image1.fpx" />
      </FPX>
   </Image>

   <Image label="My Second Image">
      <FPX>
         <SRC value="image2.fpx" />
         <Width value="8096" />
         <Height value="12300" />
      </FPX>
      <Options>
         <NoNav value="true" />
      </Options>
   </Image>
   ...
</Images>
...
```

3. **Retrieving automatically generated image lists from a server**

Example – Complete XML configurations

```
...
<Plugins>
   <Plugin src="showcase" server="..." dir="..." />
   ...
</Plugins>
...
```

Method 1 and **Method 2** can be freely combined while **Method 3** retrieves a complete image list ignoring previously defined <images> nodes.

Image Order

The order of the images within the thumbnail bar corresponds to the order of the <image> declarations within the <images> section.

Graphical Bookmarks and Product Tours

The Showcase does not re-load images if the image source did not change (e.g. "FXPSrc" and "Effects"). You can therefore use the Showcase to create graphical bookmarks or product tours by specifying the same image source ("FPXSrc") for multiple images and defining different "InitialView" parameters. The Showcase parameter "UseInitialView" should be set to "true" (default) in this case.

FSI Parameters and FSI Showcase

All FSI parameters and FSI Plug-in parameters can be used with FSI Showcase Add-on. The only exception is the plug-in integration which will be described later on. The Showcase Add-on follows the hierarchy displayed below to build the configuration for an image:

Hierarchy of parameter definitions

1. Default Configuration File
2. Showcase Configuration File
3. Image Configuration File
4. Image Node
5. HTTP Query

This way you can define global parameters for all images by defining the parameters in
1 (default configuration)
2 (showcase configuration) or
5 by HTTP query.

Parameters specific to certain images have to be defined in the external image configuration file (**3**) or in the <image> node of the image list (**4**).

Due to the hierarchy displayed above parameters declared by query supersede all other parameter definitions. This can be especially useful if you want to supersede parameters defined in external configuration files, for example superseding the "MenuAlign" parameter.

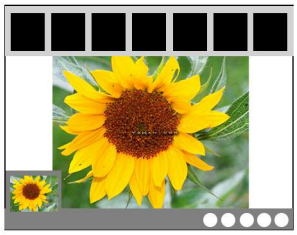
Defining the Showcase Layout

The appearance of the Showcase Add-on can be customized by a large number of configuration parameters. The parameters will be described in detail in the sections "Basic Parameters" and "Advanced Parameters".
This section describes general layout considerations and measures.

Parameter Layout:

Outside	Inside	Floating
Shrinks the size of the viewer by the initial size of the thumbnail bar	Shrinks the size of the image (view port) by the initial size of the thumbnail bar	Floats above the viewer without modifying the viewer or image size

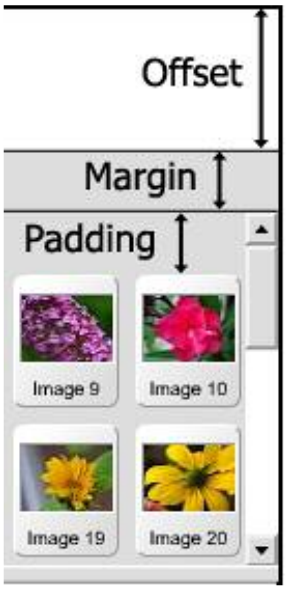
Parameter Align

Right	Left	Top	Bottom
			

The figures above show the alignment with Layout set to "Outside" and "MenuAlign set to "BL" (Bottom Left). You can achieve various layouts by modifying the 2 Showcase parameters ("**Layout**" and "**Align**") and the Viewer parameter "**MenuAlign**".

In addition to the layout parameters explained above you can define **Offsets, Margins and Paddings** for each side of the Showcase. If you use the "Inside" or "Floating" layout you will most likely have to define an offset to avoid parts of the Showcase being hidden below the menu bar.

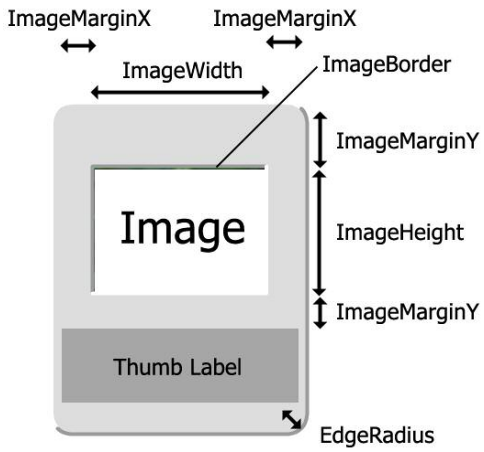
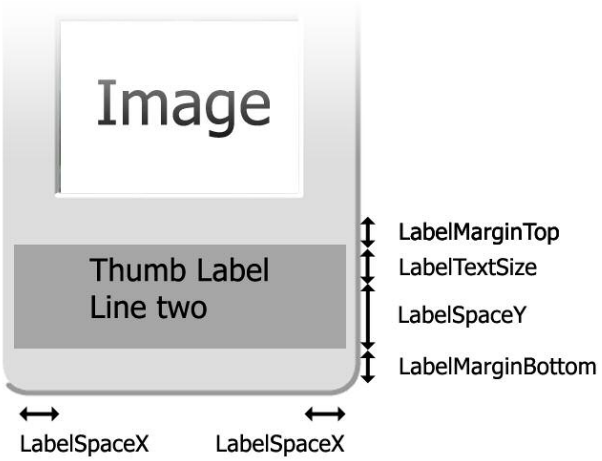
The actual values depend on the FSI Skin being used. The figure to the right illustrates the parameters "**OffsetTop**", "**MarginTop**" and "**PaddingTop**".
The values for the right, left and bottom side can be set accordingly.



- Use **Offset** to offset the entire thumbnail bar
- Use **Margin** to offset the scroll bar and thumbnails
- Use **Padding** to offset the thumbnails

Tip:
Using Offset values you can define small thumbnail bars that do not occupy the entire width or height of the viewer.

The chapter "Advanced Parameters" provides a description of parameters which can be used to resize and/or colorize all elements of the thumbnail framework. If you just want to change the general color scheme you can use the parameter "**BaseColor**" instead. You can rescale the thumbnail using the parameter "**BaseScale**".

Thumbnail Framework	Thumbnail Labels
	

Using FSI Showcase and FSI plug-ins

You can use additional plug-ins just like with FSI Viewer. There are some rules you should follow when working with additional plug-ins:

- You can not add plug-ins dynamically after FSI Showcase loaded.
  You can on the other hand disable plug-ins for individual images adding an <options> parameter with the name of the plug-in to the corresponding <image> tags of the image list:

Example – Disabling a plug-in inside an <image> definition

```
...
<Images>
   ...
   <Image label="No Magnifier" >
      <FPX>
         <SRC value="image2.fpx" />
         <Width value="8096" />
         <Height value="12300" />
      </FPX>
      <Options>
         <Magnifier value="false" />
      </Options>
   </Image>
   ...
</Images>
...
```

- Plug-in parameters defining the initial state of a plug-in do only apply to the first image. The initial state does not get restored when selecting an image in the thumbnail bar. For example if you set the "visible" parameter of the "Magnifier" plug-in to "true" it will be initially visible, but it will not be displayed each time the user selects an image if the user disabled the magnifier glass by clicking the corresponding button.

Showcase Parameters

The Showcase parameters described in the following can be provided in 3 different ways:

a) Attributes of the plug-in tag

Example – Parameter provided as attribute

```
...
<Plugins>
   <Plugin src="showcase" Layout="Outside" />
   ...
</Plugins>
...
```

b) Child nodes of the plug-in node

Example – Parameter provided as child nodes

```
...
<Plugins>
   <Plugin src="showcase" >
      <Layout value="Outside" />
      <Align value="Left" />
   </Plugin>
   ...
</Plugins>
...
```

c) By HTTP query value prefixed with "showcase_"

Example – Parameter provided by query

```
...
<Param name="movie" value="fsi.swf?showcase_layout=Outside">
...
```

The following types of values can be entered:

Type	Example
Number	"90"
Pixel	"90"
String	"ZoomIn"
URL	"http://www.neptunelabs.com/"
Bool	either "0" / "1" or "true" / "false"
HexColor	"FF00FF"
Color Transition	"FF00FF, FFFFFF"

Alphabetical Index of FSI Showcase Parameters

Align 70
Background 76
BackgroundAlpha 76
BaseColor 69
BaseScale 69
Dir 74
DragBar 71
DragMenu 71
EdgeRadius 79
HoverRotation 83
HoverZoom 83
HScroll 70
ImageBorderHighlight 78
ImageBorderShadow 78
ImageBorderWidth 77
ImageHeight 77
ImageMarginX 77
ImageMarginY 77
ImageWidth 77
InitialImage 72
InitialSize 70
LabelMarginBottom 85
LabelMarginTop 85
LabelMode 83
LabelOffsetTop 85
LabelSpaceX 84
LabelSpaceY 84
LabelTextSize 83
Layout 69
MarginBottom 81
MarginLeft 81
MarginRight 81
MarginTop 81
MinimizeOnSelect 72
OffsetBottom 80
OffsetLeft 80
OffsetRight 80
OffsetTop 80
PaddingBottom 82
PaddingLeft 82
PaddingRight 82
PaddingTop 82
PreloadThumbs 85
Query 74
ScrollArrow 77
ScrollBaseColor 76
ScrollTrack 76
Server 75
TextOutsideFrame 72
ThumbActiveFace 78
ThumbBorderActiveHighlight 79
ThumbBorderActiveShadow 79
ThumbBorderActiveWidth 78
ThumbBorderHighlight 79
ThumbBorderSelectedHighlight 79
ThumbBorderSelectedShadow 79
ThumbBorderSelectedWidth 78
ThumbBorderShadow 79
ThumbBorderWidth 78
ThumbEffects 73
ThumbFace 78
ThumbSelectedFace 78
ToolTips 72
UseInitialView 71

Basic Parameters

BaseColor	
Description:	Base color of the thumbnail bar
Syntax:	HexColor
Default:	"DDDDDD" (light gray)
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Base color for the thumbnail frame work, the scroll bar and the background of the thumbnail bar. All color values will automatically be derived from this color if not defined separately (see "Advanced Parameters").

BaseScale	
Description:	Scale of thumbnail images
Syntax:	Number
Default:	100
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Scale of thumbnails in percent, e.g. "150" for 150% of the original size.
The user can additionally scale the thumbnails by clicking the corresponding buttons in the drag bar menu. Please note that this value does not change the dimension of thumbnails being retrieved from the server (see ImageWidth and ImageHeight).

Layout	
Description:	Defines the way FSI Showcase integrates into the viewer
Syntax:	"outside", "inside" or "floating"
Default:	"outside"
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

- **Outside** (default)
  The showcase occupies the entire height or width of the viewer instance shrinking the viewer size by the initial size of the thumbnail bar. The thumbnail bar will be placed **above** all viewer elements when enlarging the thumb bar by default.
- **Inside**
  The showcase reduces the view port of the viewer by the initial size of the thumbnail bar, leaving the menu elements unchanged. The thumbnail bar will be placed **below** all viewer elements when enlarging the thumb bar by default.

- **Floating**
  The showcase floats above the view port without modifying the viewer nor the view port size. The thumbnail bar will be placed **below** all viewer elements when enlarging the thumb bar by default.

Align	
Description:	Position of the thumbnail bar
Syntax:	"left", "top", "right", "bottom"
Default:	"right"
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the position of the thumbnail bar.
Please note that the orientation of a possible scrollbar can be defined by the parameter "**HScroll**".

InitialSize	
Description:	Initial size of the thumbnail bar
Syntax:	Size in pixel or percent or in number of items
Default:	"1 item"
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the initial size of the thumbnail bar.
You can define the size in absolute pixel values (e.g. "120"), in percent of the viewer size ("50%") or by the number of thumbnails being displayed horizontally (Align="left"/"right") or vertically (Align="top"/"bottom").

If you set the "Layout" parameter to "Outside" or "Inside" this value additionally defines the minimum size of the thumbnail bar the user can achieve by dragging the thumbnail bar.

HScroll	
Description:	Display a horizontal scroll bar if required
Syntax:	Boolean
Default:	False
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

- **False**
  Arrange thumbnails vertically and use a **vertical** scroll bar if required

- **True**
  Arrange thumbnails horizontally and use a **horizontal** scroll bar if required

DragBar	
Description:	Enable/Disable dynamic resizing of the thumbnail bar
Syntax:	"Resizable", "Fixed" or "Hidden"
Default:	"Resizable"
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

- **Resizeable**
  The user **is** able to resize the thumbnail bar dynamically.
  The width of the splitter bar of the showcase is increased.
- **Fixed**
  The user **is not** able to resize the thumbnail bar dynamically.
  The width of the splitter bar of the showcase is normal.
- **Hidden**
  The user **is not** able to resize the thumbnail bar dynamically and the splitter bar of the showcase is invisible.

DragMenu	
Description:	Show/Hide the buttons in the thumbnail bar
Syntax:	Boolean
Default:	True
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Show the buttons of the thumbnail bar enabling the user to resize the thumbnails and the bar itself. This parameter is always "false" if the parameter "DragBar" is "hidden".

UseInitialView	
Description:	Use InitialView parameters to for thumbnail images
Syntax:	Boolean
Default:	True
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Use the image section defined by the parameter "InitialView" to create the thumbnail image if set to "true". If set to "False" the entire image will be used.

InitialImage	
Description:	Image selected at startup
Syntax:	Number
Default:	1
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

The index of the image selected on startup. Enter the index of the image from 1 to *n* images.

ToolTips	
Description:	Show tool tips when hovering above a thumbnail
Syntax:	Boolean
Default:	True
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Show tool tips when the mouse cursor hovers above a thumbnail. The text presented in the tool tip is the text of the thumbnail.
This option can be useful if you choose to hide the text on the thumbnail or if the thumbnail label is too long to be entirely displayed on the thumbnail itself. Multi-line texts are being presented separated by commas.

MinimizeOnSelect	
Description:	Minimize the thumbnail bar on select
Syntax:	Boolean
Default:	True
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Minimize the thumbnail bar to the initial size when the user selects (clicks) a thumbnail.

TextOutsideFrame	
Description:	Display text outside thumbnail frame
Syntax:	Boolean
Default:	False
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Display the thumbnail label inside the thumbnail frame work (false) or outside the frame work (true). The option only applies to thumbnails in normal state in does not have an impact on active and selected thumbnails.

ThumbEffects	
Description:	Image manipulation parameters applied to thumbnail images
Syntax:	String
Default:	---
Context:	<Options>
Version:	3.1.1 or higher

Applies image manipulation effects like "sharpen" or "quality" to all thumbnail images. Please refer to the → FSI Viewer "effects" parameter for a detailed description.

ImageList Parameters

If you are using eRez imaging server version 3 or above with FSI Showcase you can additionally use automatically generated showcase configuration files for a specific directory on the server or according to a search query.
The generated showcase configuration contains all required data listing images in alphabetical order.

To modify showcase parameters (e.g. ThumbWidth) you can pass the parameters by HTTP query attached to the Showcase URL (→FSI Parameters).

Experienced users might additionally edit the server template files used to generate the image list. The corresponding file is located in the 'WEB-INF/templates/fsi/' directory of your eRez imaging server:

Filename	Description
image_list.xml	Creates the <Images> section of the Showcase configuration

The following parameters have to be passed to the Showcase in order to use automatic showcase configurations:

Dir	
Description:	eRez image share (eRez only)
Syntax:	String
Default:	---
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Specifies the image directory on the eRez Imaging Server the <Images> section will be created for, e.g. 'images/paintings/'

Query	
Description:	Search query (eRez server only)
Syntax:	String
Default:	---
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Specifies a search query to be used to generate the showcase, e.g. "foo*".
Can be used alternatively to the "dir" parameter.

Server	
Description:	Path to imaging server (eRez server only)
Syntax:	String
Default:	---
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Specifies the URL of the eRez imaging server to generate the configuration.
E.g. 'http://mydomain.com/erez2/'

This parameter can be omitted if the Showcase has been installed to the default location on the eRez imaging server.

Please note: Loading configurations across domain boundaries will fail due to a security restriction of the Macromedia Flash™ plug-in.

Advanced Parameters

The following parameters can be used to customize the appearance of the thumbnails and the thumbnail bar. Please note that you can easily change the color scheme by defining a **BaseColor** rather than defining all colors individually. To change the overall scale of thumbnails you can use the basic parameter "BaseScale".

Background	
Description:	Color of the background of the thumbnail bar
Syntax:	HexColor
Default:	Defined by parameter BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Color of the background of the thumbnail bar.

BackgroundAlpha	
Description:	Opacity of the background of the thumbnail bar
Syntax:	Number
Default:	100
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the opacity of the thumbnail bar background from "0" (invisible) to "100" (opaque).

ScrollBaseColor	
Description:	Base color of the scroll bar
Syntax:	HexColor
Default:	Derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the scroll bar.

ScrollTrack	
Description:	Color of the track bar of the scroll bar
Syntax:	HexColor
Default:	Derived from ScrollBaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the track area of the scroll bar.

ScrollArrow	
Description:	Color of the arrows of the scroll bar
Syntax:	HexColor
Default:	Derived from ScrollBaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a color for the arrows on the buttons of the scroll bar.
Please note that the same color will be used for the button labels of the drag bar menu.

ImageWidth and ImageHeight	
Description:	Dimension of thumbnail images
Syntax:	Pixel
Default:	45
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the maximum physical dimension of thumbnail images in pixel. The actual size of thumbnail images depends on the aspect ratio of an image.

ImageMarginX and ImageMarginY	
Description:	Space around thumbnail image
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Space between image border and thumbnail border left/right of the image (ImageMarginX) and above/below the image (ImageMarginY).
If you choose round edges (parameter "EdgeRadius") this value will automatically be increased if required to ensure that the image does not overlap the thumbnail edge.

ImageBorderWidth	
Description:	Thickness of the border around the thumbnail image
Syntax:	Pixel
Default:	1
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the thickness of the border around the thumbnail image.
Setting this parameter to "0" disables the image border.

ImageBorderHighlight	
Description:	Highlight color of the border around the thumbnail image
Syntax:	HexColor
Default:	Derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the highlighted section of the image border (left and top).

ImageBorderShadow	
Description:	Shadow color of the border around the thumbnail image
Syntax:	HexColor
Default:	Derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the darkened section of the image border (right and bottom).

ThumbFace	
Description:	Color of the thumbnail face
Syntax:	HexColor or color transition
Default:	Color transition derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the thumb face.
This value can be either defined as a solid color (e.g. "00FF00") or as a linear color transition from top/left to bottom/right by entering two color values separated by comma (e.g "FFAA00, 00FF00").

The corresponding parameters for active and selected state are:
ThumbActiveFace and **ThumbSelectedFace.**

ThumbBorderWidth	
Description:	Thickness of the thumbnail border
Syntax:	Pixel
Default:	1
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Thickness of the border around the thumbnail. Enter "0" to disable the border.

The corresponding parameters for active and selected state are:
ThumbBorderActiveWidth and **ThumbBorderSelectedWidth.**

ThumbBorderHighlight	
Description:	Highlight color of the thumbnail border
Syntax:	HexColor
Default:	Color derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the highlighted section of the thumbnail border (left and top).

The corresponding parameters for active and selected state are:
ThumbBorderActiveHighlight and **ThumbBorderSelectedHighlight.**

ThumbBorderShadow	
Description:	Shadow color of the thumbnail border
Syntax:	HexColor
Default:	Color derived from BaseColor
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the color of the darkened section of the thumbnail border (right and bottom).

The corresponding parameters for active and selected state are:
ThumbBorderActiveShadow and **ThumbBorderSelectedShadow.**

EdgeRadius	
Description:	Radius of rounded thumbnail edges
Syntax:	Pixel
Default:	6
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines a radius for drawing rounded edges of the thumbnail frame.
Use "0" for straight edges and positive values for rounded edges.

OffsetTop	
Description:	Offset from the top of the viewer instance
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the offset from the top of the viewer instance to the top of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

OffsetBottom	
Description:	Offset from the bottom of the viewer instance
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the offset from the bottom of the viewer instance to the bottom of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

OffsetLeft	
Description:	Offset from the left of the viewer instance
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the offset from the left of the viewer instance to the left of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

OffsetRight	
Description:	Offset from the right of the viewer instance
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the offset from the right of the viewer instance to the right of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

MarginTop	
Description:	Margin from the top of the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the margin in pixel from the top of the thumbnail bar to the content and scrollbar of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

MarginBottom	
Description:	Margin from the bottom of the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the margin in pixel from the bottom of the thumbnail bar to the content and scrollbar of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

MarginLeft	
Description:	Margin from the left of the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the margin in pixel from the left of the thumbnail bar to the content and scrollbar of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

MarginRight	
Description:	Margin from the Right of the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the margin in pixel from the right of the thumbnail bar to the content and scrollbar of the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

PaddingTop	
Description:	Padding of thumbnails inside the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the top padding of thumbnails inside the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

PaddingBottom	
Description:	Padding of thumbnails inside the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the bottom padding of thumbnails inside the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

PaddingLeft	
Description:	Padding of thumbnails inside the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the left padding of thumbnails inside the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

PaddingRight	
Description:	Padding of thumbnails inside the thumbnail bar
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

This value defines the right padding of thumbnails inside the thumbnail bar of the showcase.
Please see the "**Showcase measurements**" figures for details.

HoverZoom	
Description:	Zoom thumbnails "onMouseOver"
Syntax:	Number
Default:	100 (disabled)
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Zoom thumbnails when the user points at a thumbnail.
The default value is "100" (no effect).
Possible values are 100% (no effect) to 250%.

HoverRotation	
Description:	Rotate thumbnails "onMouseOver"
Syntax:	Number
Default:	0 (disabled)
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Rotate thumbnails when the user points at a thumbnail.
The default value is "0" (no effect).
Possible values are 0 (degree) to 360 (degree).

The following parameters can be used to customize the text displayed on the thumbnails.

LabelTextSize	
Description:	Text size of the thumbnail labels
Syntax:	Pixel
Default:	0 (no text)
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Text size in pixel for the text presented below the thumbnail image.
Use "0" to disable text and positive values (e.g. "8") to enable texts on the thumbnails.

LabelMode	
Description:	Content of the thumbnail labels
Syntax:	Number
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

Defines the content of the text presented below the thumbnail image.
The following modes are available:

- **Mode "0"** (default)
  Display the text defined by "Label" parameter.
- **Mode "1"**
  Display the index of the image.
- **Mode "2"**
  Display the index of the image plus the text defined by "Label" parameter.
- **Mode "3"**
  Display the prefix only.

LabelPrefix	
Description:	Prefix applied to thumbnail texts
Syntax:	String
Default:	---
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

A prefix string applied to all thumbnail labels or "false" for now prefix.
This is especially useful with "LabelMode 2" and "LabelMode 3" to display the image index with a prefix like "#" or "image " or if you want to display a general tool tip for all images like "Click to view image".

LabelSpaceX	
Description:	Additional horizontal space for thumbnail text
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to increase or decrease the width of the thumbnail text area. Increasing the width might display the text across the thumbnail boundaries or even lead to overlapping thumbnail texts.

LabelSpaceY	
Description:	Additional vertical space for thumbnail text
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to increase or decrease the height of the thumbnail text area. If you want to use multiple lines of text you can increase the height of the text area by the following rule:

LabelSpaceY = AddtionalLines x LabelTextSize

If the thumbnail text is too long to be displayed in the text area you can either increase this value or enable tool tips (parameter "ToolTips).

LabelMarginTop and LabelMarginBottom	
Description:	Margin above/below the thumbnail text
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to add additional space above (LabelMarginTop) or below (LabelMarginBottom) the thumbnail text.

LabelOffsetTop	
Description:	Adjust vertical position of the thumbnail text
Syntax:	Pixel
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

You can use this value to adjust the vertical position of the text area of the thumbnail. Negative values move the text area up while positive values move the text area down.

PreloadThumbs	
Description:	Load all thumbnail images on startup
Syntax:	Number
Default:	0
Context:	Showcase plug-in node or HTTP query prefixed with "showcase_"

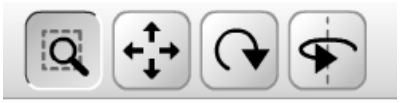
Setting the value to "0" makes the showcase load thumbnails on-demand.
Setting positive values enables the pre-loader which loads all images on startup The given value specifies the number of images to preload simultaneously.

Please note: Be careful when activating this option especially with automatically generated image lists or large image lists as it might cause higher server loads and increased (and possibly unnecessary) network traffic.

Plug-in Reference

NeptuneLabs FSI Viewer Plug-in
Plug-in Mousemodes

Plug-in Target:
User interface



Plug-in Location:
/plugins/mousemodes.plg

Function:
Integrates buttons into the menu bar to select the mouse mode.

Syntax:
```
<Plugin src="mousemodes" />
```

The "Mousemodes" plug-in provides 3 buttons (2D images) or 4 buttons (3D presentations) to select the current mouse mode.

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu buttons
Syntax:	Number
Default:	0
Context:	Plug-in Tag

Specifies the space in pixel left of the menu button(s) of the plug-in.

Mode *n*	
Description:	Removes the button for mouse mode *n* from the menu bar
Syntax:	Bool
Default:	---
Context:	Plug-in Tag

Provides the possibility to hide specific mouse mode buttons.

Example - Remove Drag-Button

```
<Plugin src="mousemodes" Mode1="false" />
```

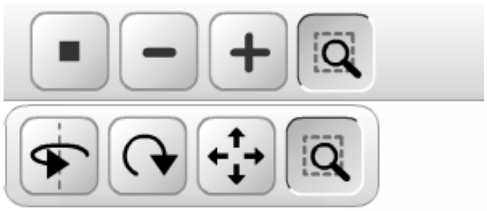
The example above shows how to hide the button for mouse mode 1 (Drag).

Mode n	Mouse Mode
Mode0	Zoom
Mode1	Drag
Mode2	Rotate (X/Y Axis)
Mode3	Rotate (Z Axis)

NeptuneLabs FSI Viewer Plug-in
Plug-in Mousemode Select

Plug-in Target:
User interface

Plug-in Location:
/plugins/mousemodeselect.plg



Function:
Integrates a button and a dropdown menu into the menu bar to select the mouse mode.

Syntax:
```
<Plugin src="mousemodeselect" />
```

The "Mousemode Select" plug-in adds a button and a dropdown menu to the menu bar to select the mouse mode.

*Some plug-ins (e.g. plug-in "Measure") add additional mouse mode buttons to the menu bar. If you include these plug-ins **after** including the MouseModes plug-in, these additional buttons will be added to the MouseModeSelect submenu.*

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in Tag

Specifies the space in pixel left of the menu button of the plug-in.

Mode *n*	
Description:	Removes the button for mouse mode *n* from the menu
Syntax:	Bool
Default:	---
Context:	Plug-in attributes

Provides the possibility to hide specific mouse mode buttons in the dropdown menu.

Example - Remove Drag-Button

```
<Plugin src="mousemodes" Mode1="false" />
```

The example above shows how to hide the button for mouse mode 1 (Drag).

Mode n	Mouse Mode
Mode0	Zoom
Mode1	Drag
Mode2	Rotate (X/Y Axis)
Mode3	Rotate (Z Axis)

NeptuneLabs FSI Viewer Plug-in
Plug-in Zoom Meter

Plug-in Target:
User interface

Plug-in Location:
/plugins/zoommeter.plg

Function:
Displays the current level of magnification.

Syntax:
```
<Plugin src="zoommeter" color="000000" />
```

The "ZoomMeter" plug-in extends the survey window with a display of the current magnification.

Plug-in Parameters

Color	
Description:	Text color
Syntax:	String
Default:	"000000" (black)
Context:	Plug-in attribute

Text color of the zoom display.
The color has to be defined as a 6-digit hexadecimal number.
(E.g. "FFFF00" for yellow).

SrcRelative	
Description:	Display magnification relative to source image or viewer size
Syntax:	Boolean
Default:	False
Context:	Plug-in attribute
Version:	3.0.0 or higher

Display the current magnification in percent of the viewer size (false) or relative to the size of the source image.

NeptuneLabs FSI Viewer Plug-in
Plug-in Max Zoom

Plug-in Target:
User interface

Plug-in Location:
/plugins/maxzoom.plg

Function:
Zoom to physical resolution of the image.

Syntax:
```
<Plugin src="maxzoom" />
```

The "MaxZoom" plug-in expands the user interface by a button to quickly zoom to the physical resolution of the source image.

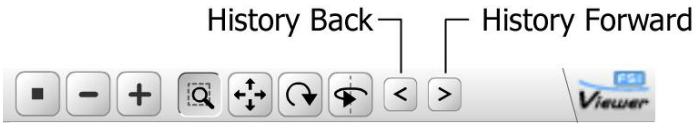
Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Specifies the space in pixel left of the menu button of the plug-in.

NeptuneLabs FSI Viewer Plug-in
Plug-in History

Plug-in Target:
User interface



Plug-in Location:
/plugins/history.plg

Function:
History functionality to jump to previously viewed image sections

Syntax:
```
<Plugin src="history" length="25" />
```

The "History" plug-in extends the user interface by 2 buttons, "History Back" and "History Forward".
Using these two buttons the user is able to navigate through image sections previously viewed.

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu buttons
Syntax:	Number
Default:	0
Context:	Plug-in Tag

Specifies the space in pixel left of the menu button(s) of the plug-in.

Length	
Description:	Length of the history list
Syntax:	Number
Default:	10
Context:	Plug-in tag

The number of image sections that will be stored in the view history from 1 to 99. If the list exceeds the amount of entries defined by the "length" parameter, the first entry of the list will be replaced.

Loop	
Description:	Loop the history list
Syntax:	Bool
Default:	false
Context:	Plug-in tag

If this value is set to "true" the viewer will display the first viewed section when exceeding the last entry of the history and the last entry when stepping before the first entry of the history list.

Providing the default value "false", the history will not be looped, this is the history list stops at the first and the last position.

SmallButtons	
Description:	Size of menu buttons
Syntax:	Bool
Default:	true
Context:	Plug-in tag

If this value is set to "false" the buttons of this plug-in will be displayed in the default button size. Otherwise the buttons appear slightly smaller than other buttons.

NeptuneLabs FSI Viewer Plug-in
Plug-in Resize

Plug-in Target:
User interface

Plug-in Location:
/plugins/resize.plg

Function:
Rearranges user interface elements after resizing the viewer

Requirements:
Flash MX or higher

Syntax:
```
<Plugin src="resize" MaxWidth="800" MaxHeight="600"/>
```

Using the resize plug-in you can define the viewer dimension in percent of your HTML document. After resizing the plug-in rearranges the user interface according to the size of the object tag without reloading the entire viewer instance and without the need for client side scripting.

Plug-in Parameters

MaxWidth and MaxHeight	
Description:	Maximum width and height
Syntax:	Number in pixel
Default:	---
Context:	Plug-in Tag

FSI Viewer can not be resized to a dimension greater than specified by these values.

MinWidth and MinHeight	
Description:	Minimum width and height
Syntax:	Number in pixel
Default:	---
Context:	Plug-in Tag

FSI Viewer can not be resized to a dimension smaller than specified by these values.

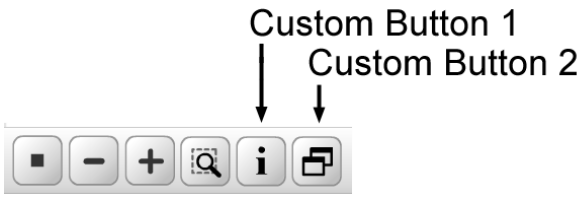
EnlargeBy	
Description:	Resize viewer in steps of *n* pixels
Syntax:	Number
Default:	1
Context:	Plug-in tag

Specifies the steps FSI Viewer will be resized by. Entering 1 means that the viewer resizes exactly according to the size of the object tag. Entering greater numbers forces the viewer to resize by the number of pixel specified. Using values > 1 increases the chance of imaging server cache hits and might thus reduce server load.

NeptuneLabs FSI Viewer Plug-in
Plug-in CustomButton

Plug-in Target:
User interface

Plug-in Location:
/plugins/custombutton.plg



Function:
Adds custom buttons to the menu bar

Syntax:
```
<Plugin src="CustomButton" buttons="btn1,btn2 />
```

The "CustomButton" plug-in can be used to integrate one or more custom buttons into the menu bar of FSI viewer. Clicking a button invokes the URL specified in the plug-in parameter. This way you can integrate buttons performing a JavaScript call or opening up a web page.

Plug-in Parameters

MenuOffset	
Description:	Offset in pixel to previous menu elements
Syntax:	Number in pixel
Default:	0
Context:	Plug-in Tag

Defines an offset of all custom buttons in pixels to elements of the menu bar.

Buttons	
Description:	Comma separated list of unique identifiers
Syntax:	String
Default:	---
Context:	Plug-in Tag

Enter a list of unique identifiers separated by "," commas. Each identifier represents a new button that can be configured using the parameters below. E.g. "`btn1,btn2`" adds two custom buttons. To specify parameters for these buttons you have to prefix the parameters by the unique identifier plus, e.g. "`btn1.Url`" defines the Url to use for the button named "`btn1`".

[ID].offset	
Description:	Offset of the button in pixels to previous (custom) buttons
Syntax:	Number in pixel
Default:	0
Context:	Plug-in Tag

Specifies an offset for the button named "[**ID**]" in pixel to previous (custom) buttons.

[ID].Url	
Description:	Url to open on click
Syntax:	String
Default:	---
Context:	Plug-in Tag

Specifies an URL to open when the user clicks the button named "[**ID**]".
You can specify a URL to a web page or alternatively a script call,
e.g. "`javascript: void alert("foo");` "
Please note: You can use →templates to be replaced by FSI Viewer at runtime.
The button appears disabled if no URL is specified.

[ID].Frame	
Description:	HTML frame to open the specified URL in
Syntax:	String
Default:	_self
Context:	Plug-in Tag

Specifies the HTML frame to open the specified Url in for the button named "[**ID**]". Besides using an identifier of a frame inside your HTML frame set you can as well use one of the predefined values "`_self`", "`_top`", "`_blank`" or "`_parent`".

[ID].ToolTip	
Description:	Tool tip text
Syntax:	String
Default:	---
Context:	Plug-in Tag

Specifies the tool tip to be displayed when the user places the cursor above the custom button named "[**ID**]".

[ID].LabelFrame	
Description:	Symbol displayed on the button
Syntax:	Number
Default:	1
Context:	Plug-in Tag

Specifies the symbol to be displayed on top of the custom button named "[**ID**]". The predefined symbols are:

GFX	ID	Example
	1	Information
	2	Close
	3	Resize or Restore
	4	Open window
	5	Print

Additionally you can create your own graphics that can be used as a label for custom buttons. To create new graphics you have to modify the file "**custombutton_gfx.fla**" located in the "/plugins" directory of your FSI setup directory with Macromedia Flash (or any compatible) editor:

1. Open the file "/plugins/fla/custombutton_gfx.fla" in your Flash editor
2. Choose an existing or create a new key frame in the main timeline
3. Add your graphic to Layer 1 ("GFX")
4. Compile, save and upload the modified file
5. Use the index of the key frame (step 2) as the "LabelFrame" parameter for your custom button

Please note:

- Make sure to upload the modified file properly after editing
- You should not modify the Flash file except for adding your custom graphic(s)
- The dimension of your graphics must match your FSI Skin. Usually the dimension should not exceed **18 x 18** pixels.

Using Templates in URL parameters

Starting from FSI version 3.2.3 you can use templates when defining URLs for custom buttons. These templates will be replaced by FSI Viewer when the user clicks a custom button. This way you can retrieve parameters from FSI Viewer dynamically and pass the values to a JavaScript function or a server side script.

Example - Templates in CustomButton URLs

```
<Plugin src="custombutton" buttons="btn1"
btn1.URL="javascript:alert('$$cfg.FPXWidth$$ x $$cfg.FPXHeight$$')" />
```

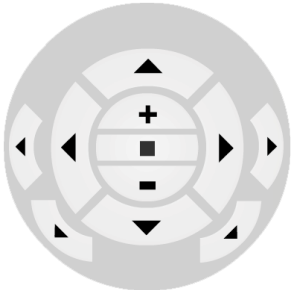
The example above outputs the size of the source image via JavaScript.
The following templates are available:

Template in URL	Replaced by…
`$$cfg.[Name]$$`	The value of the FSI Viewer parameter *[name]*. Be sure to use the long parameter names only (e.g. "`FPXWidth`" instead of "`Width`"). You can retrieve parameter values retrieved from the imaging server using this template as well. E.g. "`$$cfg.FPXWidth$$`" returns the size of the source image – whether specified manually or retrieved from the imaging server.
`$$View$$`	A string representing the current image section in FSI Viewer (see Parameter →"**InitialView**" for details)
`$$URLView$$`	The complete URL to the image currently displayed in FSI Viewer. The resulting string contains the templates "`$$width$$`", "`$$height$$`" and "`$$template$$`" (eRez only) which need to be replaced in JavaScript or server side script.
`$$URLScene$$`	Same as above, but representing the entire scene image.
`$DownloadRatio$$`	Aspect ratio representing the factor "width/heigth". You can use this value to calculate the width of the image to download.
`$$Zoom$$`	Current magnification relative to the initial size of the image.
`$$Selection$$`	A string representing the current selection (**requires plug-in SelectFrame**)
`$$Measure$$`	A string representing the current measurement (**requires plug-in Measure**)

NeptuneLabs FSI Viewer Plug-in
Plug-in NavExtension

Plug-in Target:
User interface

Plug-in Location:
/plugins/navextension.plg



Function:
Additional navigation control.

Syntax:
```
<Plugin src="navextension" visible="true" />
```

The "NavExtension" plug-in adds a dragable navigation control to the FSI Viewer user interface that can be hidden and shown with a corresponding button in the menu bar. The control automatically adds buttons for Scenes and scene sets depending on the image configuration.

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attributes

Specifies the space in pixel left of the menu button of the plug-in.

Visible	
Description:	Initial display state
Syntax:	Bool
Default:	false
Context:	Plug-in attribute

Defines the initial visibility state of the control.

X	
Description:	Initial X position in pixel
Syntax:	Number
Default:	0
Context:	Plug-in Tag

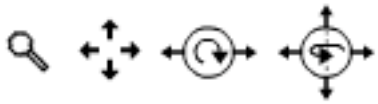
Initial horizontal position of the control in pixels relative to the top-left corner of the viewer.

Y	
Description:	Initial Y position in pixel
Syntax:	Number
Default:	0
Context:	Plug-in Tag

Initial vertical position of the control in pixels relative to the top-left corner of the viewer.

NeptuneLabs FSI Viewer Plug-in
Plug-in Software Cursor

Plug-in Target:
User interface



Plug-in Location:
/plugins/softwarecursor.plg

Function:
Adds a software cursor to the FSI Viewer.

Syntax:
```
<Plugin src="softwarecursor" />
```

The "Software Cursor" plug-in adds a software cursor representing the current mouse mode (Zoom, Drag, Rotate-2D, Rotate 3D).

Plug-in parameters

No parameters available.

NeptuneLabs FSI Viewer Plug-in
Plug-in MouseWheel

Plug-in Target:
User interface

Plug-in Location:
/plugins/mousewheel.plg

Function:
Adds mouse wheel support to FSI Viewer (Zoom, Rotate)

Syntax:
`<Plugin src="mousewheel" />`

Requirements:
MSIE 6.0, JavaScript

The "MouseWheel" plug-in enables the user to scroll or rotate an image using the mouse wheel.

Implementing MouseWheel support

1) Add a class-attribute of value "FSIViewer" to all FSI Viewer Object tags:
E.g. `<object class="FSIViewer" classid= ...>`

2) the following javascript to your HTML source:

JavaScript– MouseWheel

```
<script type="text/javascript">
   <!--
   FSIViewers=new Array();

   function WheelInit(){
       if (!document.all) return;

       t = document.getElementsByTagName("object");
       if (t) for (count=0,i=0;i<t.length;i++){
           if (t[i].className=="FSIViewer") {
               FSIViewers[count++]=t[i];
               t[i].SetVariable("CatchMouseWheel", false);
           }
       }
   }
```

```
   document.onmousewheel = function () {
       for (i=0;i<FSIViewers.length;i++){
           if (FSIViewers[i].GetVariable("CatchMouseWheel")=="true"){
               FSIViewers[i].SetVariable("WheelDelta",
event.wheelDelta);
               return false;
            }
       }
   }
   //-->
</script>
```

3) Add the function call "WheelInit();" to the onload event of the body tag:

E.g. `<BODY onload="WheelInit();" ...>`

4) Add the "MouseWheel" plug-in to your _default.fsi or image specific .fsi file:

Example - Adding the Mousewheel plug-in

```
<Plugins>
   <Plugin src="mousewheel" />
</Plugins>
```

MouseWheel Plug-in Parameters

You can specify 4 different actions to take place when the user uses the mousewheel:

- **Zoom** (Zoom, Default)
- **Rotate** (Rotate Z-Axis)
- **Scene** (Rotate Y-Axis)
- **SceneSet** (Rotate X-Axis)

Default	
Description:	Default mousewheel action
Syntax:	String
Default:	"Zoom"
Context:	Plug-in Tag

Defines the default action of the mouse wheel, provided that no other action has been defined using the Mode[n] parameter.

Example

```
<Plugin src="mousewheel" default="Rotate" />
```

Mode[*n*]	
Description:	MouseWheel action for MouseMode *n*
Syntax:	String
Default:	Specified by "default" parameter or "Zoom"
Context:	Plug-in attribute

Defines the action of the mouse wheel when mouse mode *n* has been activated.

Example - Defining rotate as default action

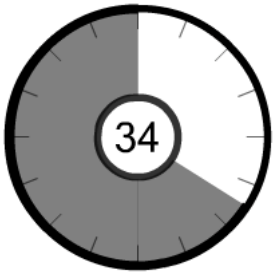
```
<Plugin src="mousewheel" default="Rotate" Mode0="Rotate" />
```

The example above shows how to specify the mouse wheel action "Rotate" for the mouse mode "Zoom".

The possible values for "n" are as following:

0 MouseMode **Zoom**
1 MouseMode **Move**
2 MouseMode **Rotate**
3 MouseMode **Tilt**

NeptuneLabs FSI Viewer Plug-in
Plug-in ClockProgress



Plug-in Target:
User interface

Plug-in Location:
/plugins/clockprogress.plg

Function:
Replacement for default progress bar

Syntax:
`<Plugin src="clockprogress" />`

The "ClockProgress" plug-in replaces the default progress bar with a round progress indicator.

Plug-in Parameters

The clockprogress plug-in can be customized by adding attributes to the <plugin> tag.
E.g. `<Plugin src="clockprogress" alpha="20"/>`

The following attributes can be used to customize the clockprogress plug-in:

Alpha	
Description:	Opacity of progress indicator
Syntax:	Number
Default:	50
Context:	Plug-in attributes

Defines the opacity of the progress indicator. Possible values range from 0 to 100.

0 transparent
100 opaque

Size	
Description:	Size of progress indicator
Syntax:	Number
Default:	60
Context:	Plug-in Tag

Defines the overall size of the progress indicator in pixel.

Color1	
Description:	Color of the area representing the amount of pending data
Syntax:	HexColor
Default:	000000
Context:	Plug-in attributes

A 6-digit hexadecimal number defining the color of the area representing the amount pending data.

Color2	
Description:	Color of the area representing the amount of data already loaded
Syntax:	HexColor
Default:	FFFFFF
Context:	Plug-in attributes

A 6-digit hexadecimal number defining the color of the area representing the amount data already loaded.

LineColor	
Description:	Line color
Syntax:	HexColor
Default:	000000
Context:	Plug-in attribute

A 6-digit hexadecimal number defining the color of the lines.

TextColor	
Description:	Text color
Syntax:	HexColor
Default:	000000
Context:	Plug-in attribute

A 6-digit hexadecimal number defining the color of the text.

posX	
Description:	Horizontal Position
Syntax:	String / Number
Default:	Depends on MenuAlign parameter
Context:	Plug-in attribute

Defines the horizontal position of the progress indicator.

Possible values:

C	Centered horizontally
Number > 0	Offset from left
Number < 0	Offset from right

posY	
Description:	Vertical position
Syntax:	String / Number
Default:	Depends on MenuAlign parameter
Context:	Plug-in attribute

Defines the vertical position of the progress indicator.

Possible values:

C	Centered vertically
Number > 0	Offset from top
Number < 0	Offset from bottom

NeptuneLabs FSI Viewer Plug-in
Plug-in Music

Plug-in Target:
User interface

Plug-in Location:
/plugins/music.plg

Function:
Adds streaming background music or sound to the FSI Viewer.

Syntax:
```
<Plugin src="music" song="music.mp3" autoplay="true" />
```

Requirements:
Flash MX or higher

The "Music" plug-in adds a background sound or music to FSI Viewer.
Additionally a button will be added to the menu bar to switch the music on or off.

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Specifies the space in pixel left of the menu button of the plug-in.

AutoPlay	
Description:	Start playing the sound on startup
Syntax:	Bool or String
Default:	false
Context:	Plug-in attribute

Set the value to Boolean "**true**" to make the sound play on startup.
From version 1.3 of the plug-in you can additionally set the value to "**onLoad**" to start the sound after image data has been loaded completely.

Song	
Description:	Path to an MPEG3 audio file (.mp3)
Syntax:	URL
Default:	---
Context:	Plug-in attribute

Defines a relative path to an MPEG3 file (*.mp3) located in the "music" subdirectory of the "/fsi" folder.

Loop	
Description:	Loop / Play the sound once
Syntax:	Bool
Default:	false
Context:	Plug-in attribute

Defines if the sound is being played once or in a continuous loop.

Volume	
Description:	Sound volume
Syntax:	Number (0-100)
Default:	---
Context:	Plug-in attribute

Defines the volume of the sound from 0 (quiet) to 100 (maximum).

KeepState	
Description:	Keep Stop/Play state when selecting images
Syntax:	Boolean
Default:	True
Context:	Plug-in attribute

For viewers with multiple images only.
Keep the playing state and ignore the "AutoPlay" parameter when switching from one image to another. This way "AutoPlay=true" does not start the sound if the user switched the button off.

NeptuneLabs FSI Viewer Plug-in
Plug-in Magnifier

Plug-in Target:
User interface

Plug-in Location:
/plugins/magnifier.plg

Function:
Adds a magnifier to the user interface

Syntax:
```
<Plugin src="magnifier" visible="true" size="10" />
```

The "Magnifier" plug-in adds a magnifier to the user interface that can be dragged using the mouse. Using the magnifier plug-in does not cause additional data traffic, as the plug-in enlarges already loaded image data by software. The magnifier can be hidden and shown using a corresponding button in the menu bar. The level of magnification and the magnifier size can be defined by plug-in parameters.

Plug-in Parameters

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Specifies the space in pixel left of the menu button of the plug-in.

Visible	
Description:	Initial state of visibility
Syntax:	Bool
Default:	false
Context:	Plug-in attribute

Defines the initial visibility of the magnifier.

Size	
Description:	Size of magnifier
Syntax:	Number
Default:	8
Context:	Plug-in attribute

Defines the overall size of the magnifier. Possible values are between 4 and 16. Please note that greater values require more CPU power.

Magnification	
Description:	Level of magnification
Syntax:	Number
Default:	2
Context:	Plug-in attribute

Defines the level of magnification where 1 means no magnification, 2 means a magnification of 200% and so on. The maximum level of magnification is 8.

Reflections	
Description:	Show or hide glass reflections
Syntax:	Bool
Default:	true
Context:	Plug-in attribute

Show (true) or hide (false) glass reflections of the magnifier.

NeptuneLabs FSI Viewer Plug-in
Plug-in HotSpots

Plug-in Target:
User Interface and Zoom Area

Plug-in Location:
/plugins/hotspots.plg

Function:
By defining clickable areas (hotspots) the user is able to interact with the image by clicking on areas inside the zoomable image.

Syntax:
```
<Plugin src="hotspots" />
```

Enterprise editions of FSI Viewer contain this plug-in by default.
For all other editions this plug-in can be optionally obtained.

The "HotSpots" plug-in adds an image map like functionality to the FSI Viewer. The plug-in adds a button to the menu bar which allows the user to show or hide the HotSpots.
The clickable areas are defined via XML inside the image specific *.fsi file and enable an increased interaction with the Viewer.

There are 4 different combinable actions that can be assigned to each hotspot:

- Display a tool tip
- Zoom to the area defined by the hotspot
- Open a HTML page
- Restart the Viewer with a different image / configuration

Adding Hotspot Functionality to your Images

Integrating the Plug-in

To integrate the HotSpot plug-in into the viewer you have to add the following line to the <Plugins> section of either the _default.fsi file or to an image specific .fsi configuration file:

```
<Plugin src="hotspots" />
```

Defining the HotSpots Section

To define HotSpots for an image you have to add an XML-section to your image specific .fsi file. The definition syntax is similar to image maps for static images inside a HTML page.
Each HotSpot is being defined by a single XML node inside the <HotSpots> group.
The basic structure of a HotSpot definition is the following:

Example – HotSpot definition

```
<HotSpots visible="true" alpha="70">
  <circle spot="" tip="" url="" />
  <rect   spot="" tip="" url="" />
</HotSpots>
```

In contrast to other main groups like e.g. "<FPX>" you can assign attributes to the HotSpot group itself. Most of these attributes define the default behavior of all included HotSpots so that you can omit these attributes in subsequent HotSpot nodes.

Global HotSpot Parameters

The attributes you can assign to the <HotSpots> group tag include:

Visible	
Description:	Initial visibility of the HotSpots
Syntax:	Bool
Default:	false
Context:	<HotSpots> group tag

Defines the initial visibility of HotSpots.
The HotSpot button state will be set accordingly.

Alpha	
Description:	Defines the opacity for all HotSpots
Syntax:	Number zwischen 0-100
Default:	100
Context:	<HotSpots> tag

Defines the opacity for all HotSpots from 0 (invisible) to 100 (opaque).

BaseURL	
Description:	Prefix for relative HotSpot URLs
Syntax:	String
Default:	---
Context:	<HotSpots> group tag

Defines a prefix that will be applied to all HotSpots containing relative URLs.
This way you can omit e.g. the domain in URL-attributes of subsequent HotSpots.

Defining Default Attributes

The following attributes define default attributes to all HotSpot that do not contain a corresponding attribute itself.
E.g. defining a **DefaultTarget** of "**_self**" will open all hyperlinks in the frame the viewer resides in, if there is no individual **Target** attribute defined.

DefaultTarget	
Description:	Default target for hyperlinks
Syntax:	String
Default:	_blank
Context:	<HotSpots> group tag

Defines the default target(frame) for HotSpot hyperlinks.
Valid parameters are "`_blank`", "`_self`", "`_top`", "`_parent`" and names of frames inside a HTML frameset.

DefaultMove	
Description:	Zoom to HotSpot area by default
Syntax:	Bool
Default:	true
Context:	<HotSpots> group tag

Defines the default value for all subsequent HotSpots that do not contain the "**Move**" attribute.

DefaultSkew	
Description:	Permit skewing of HotSpots
Syntax:	Bool
Default:	true
Context:	<HotSpots> tag

Defines the default value for all subsequent HotSpots that do not contain the "**Skew**" attribute.

DefaultColor	
Description:	Default color of HotSpot borders
Syntax:	String
Default:	FF0000
Context:	<HotSpots> tag

Defines the default color of HotSpot borders.
The value has to be a 6 digit hexadecimal number like "FFFF00" for yellow or "0000FF" for blue.

DefaultFill	
Description:	Fill the HotSpot area by default
Syntax:	Bool
Default:	true
Context:	<HotSpots> group tag

Defines the default value for all subsequent HotSpots that do not contain the "**Fill**" attribute.
If this value is set to false, all HotSpots will by shown as outlines by default.

Defining Individual HotSpots

Each HotSpot definition must at least contain the name of the HotSpot shape and the "**Spot**" attribute which defines the position of the HotSpot.
The tag name of each definition specifies the shape of the HotSpot.

Possible shapes are:

- **rect** (Square / Rectangle)
- **circle** (Circle / Ellipse)
- **star** (Star)

Example – HotSpot definition

```
<HotSpots visible="true" DefaultTarget="_blank"
BaseURL="http://Domain/" >

  <circle spot="1, 1, 0.35842, 0.15383, 0.61261, 0.40802, 0"
tip="circle />
  <rect spot="1, 10, 0.54314, 0.68446, 0.74036, 0.88168, -56"
tip="rectangle" url="rect.htm" />

</HotSpots>
```

Please refer to the chapter "**HotSpot Authoring Tips**" for useful tips on how to create HotSpots.

HotSpot Parameters

The following attributes can be assigned to each HotSpot:

Spot	
Description:	Position of the HotSpot
Syntax:	String
Default:	---
Context:	HotSpot tag

Defines the position of the HotSpot.
It is recommended to use the plug-in →**SelectFrame** to obtain this value. Please refer to the chapter "HotSpot Authoring tips" for more information on this parameter.

View	
Description:	Image section to display
Syntax:	String
Default:	Value of spot parameter
Context:	HotSpot tag
Version:	3.2.3 and above

Defines the position to display when move is set to true.
If this parameter is undefined, the value of spot will be used instead.
Please refer to the chapter "HotSpot Authoring tips" for more information on this parameter.

Tip	
Description:	Tooltip
Syntax:	String
Default:	---
Context:	HotSpot tag

Defines a string to be used for a tooltip when the user moves the cursor over the HotSpot.

Move	
Description:	Zoom to HotSpot area on click
Syntax:	Bool
Default:	false
Context:	HotSpot tag

Zoom to the image section covered by the HotSpot, if the user clicks the HotSpot.
You can use the "**DefaultMove"** parameter to change the default behavior.

Skew	
Description:	Allow skewing of the HotSpot area
Syntax:	Bool
Default:	false
Context:	HotSpot tag

If this parameter is set to false the width of the hotspot area will equal the height of the HotSpot regardless of the values provided with the "**Spot**" parameter.
You might want to use this parameter to ensure that e.g. HotSpots defined with the <circle> tag look like circles instead of ellipses.
You can use the "**DefaultSkew**" parameter to change the default behavior.

Color	
Description:	Color of the HotSpot border
Syntax:	String
Default:	FF0000
Context:	HotSpot tag

Defines the color of the HotSpot border.
The value has to be a 6 digit hexadecimal number like "FFFF00" for yellow or "0000FF" for blue. You can use the "**DefaultColor**" parameter to change the default border color.

Fill	
Description:	Fill the HotSpot area
Syntax:	Bool
Default:	false
Context:	HotSpot tag

If this value is set to false, the HotSpot will by shown as an outline.
You can use the "**DefaultFill**" parameter to change the default behavior.

URL	
Description:	Hyperlink of the HotSpot
Syntax:	URL
Default:	---
Context:	HotSpot tag

Defines a hyperlink to be opened if the user clicks the HotSpot.
The URL can be given relatively or absolutely.
The Target parameter (or respectively the "**DefaultTarget**" parameter) defines the targetframe for the hyperlink. Please refer to the "**BaseURL**" parameter as well.

Target	
Description:	Target frame for hyperlinks
Syntax:	String
Default:	_blank
Context:	HotSpot tag

Defines the targetframe for HotSpot-hyperlinks.
Valid parameters include "`_blank`", "`_self`", "`_top`", "`_parent`" and names of frames inside a HTML frameset.

Please refer to the "**DefaultTarget**" parameter for details on how to set a default target for hyperlinks.

CFG	
Description:	FSI parameters to be used on click
Syntax:	String (query)
Default:	---
Context:	HotSpot tag

If you define a CFG parameter for a HotSpot the viewer will be reinitialized with the given configuration if the user clicks the HotSpot.
This can be useful to switch from one image to another if the user clicks a HotSpot.

The syntax of the CFG parameter has to consist of FSI Parameters concatenated with the ampersand character (like a HTTP query).

```
Parameter1=value1&Parameter2=value2…
```

Example:
`cfg="cfg=image2&NoNav=1"`

Defining the parameter like this, the viewer would reinitialize using the configuration file "image2.fsi" providing the FSI parameter "NoNav" with a value of "true".

Please refer to the FSI Viewer manual for a list of all FSI Parameters and for details on how to create a query string that provides FSI parameters.

SelectImage	
Description:	Select another image (FSI Showcase only)
Syntax:	Number
Default:	---
Context:	HotSpot tag

Clicking the HotSpot selects another image inside FSI Showcase.
The index can be specified absolutely by providing a number only. The first image is being represented by "0".

Alternatively you can select an image relatively from the currently selected image. To select an image relatively you have to provide a number prefixed by "+" or "-", e.g. "+1" selects the next image.

Hotspot Authoring Tips

The following section provides tips on how to obtain the string value required to define the "**Spot**" and the optional "**View**" parameter of a HotSpot.

The Spot parameter consists of 7 numbers concatenated with commas and provides the scene set, the Scene, the image section (as a rectangle) and the rotation.

Example: 1, 10, 0.19056, 0.33579, 0.44862, 0.59385, 35

scene set **1**, Scene **10**, Rectangle [**0.19056, 0.33579, 0.44862, 0.59385**], rotation **35 °**

Although you can of manually enter the parameters, it is recommended to use the plug-in →**SelectFrame** to obtain this value.

Obtaining Hot Spot positions

Include the plug-in "SelectFrame" by adding the following parameters to the "_default.fsi" or any other configuration file.

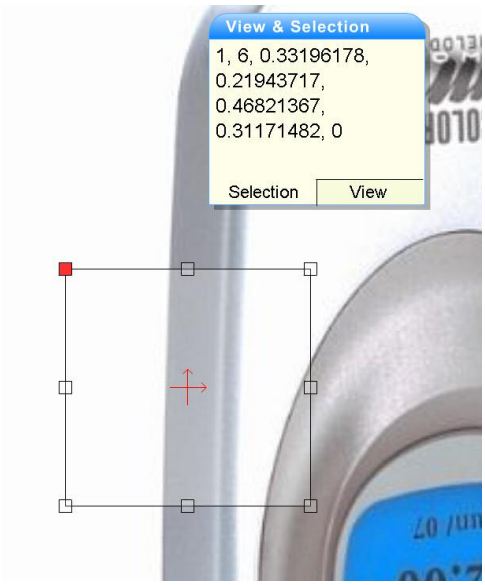
Example – Inlude plug-in SelectFrame

```
<plugins>
   <plugin src="SelectFrame" visible="1" window="1" rotate="1"
AspectRatio="1:1" />
</plugins>
```

After including the plug-in you will be able to author hotspots directly in FSI Viewer.

You can use the value displayed in the "**Selection**" window for the "**Spot**" parameter of the hotspot.

You can use the value displayed in the "**View**" window for the optional "**View**" parameter.



When modifying the select box please use:

SHIFT to lock the aspect ratio of the selection
CTRL to resize the selection relative to the center
ALT to rotate the selection
ALT + SHIFT to rotate the selection in steps of 5 degrees

HotSpot Z-Order

When defining Hot Spots that overlap each other, you have to keep the z-order of the Hot Spots in mind. The z-order of Hot Spots is determined by the sequence in which they are defined in the configuration file, from bottom to top.

As a rule of thumb you should define big Hot Spot areas first to ensure that smaller Hot Spots are not completely hidden underneath.

NeptuneLabs FSI Viewer Plug-in
Plug-in JavaScript Bridge

Plug-in Target:
Scripting

Plug-in Location:
/plugins/jsbridge.plg

Function:
Integrates a JavaScript interface to enable interaction between FSI Viewer and JavaScript contained in the hosting HTML page.

Syntax:
```
<Plugin src="jsbridge" />
```

Requirements:
Browser with live connect support between Flash and JavaScript (Livewire)

This plug-in can be optionally obtained.
Using the plug-in without a valid licence key activates the evaluation mode.

The "JavaScript Bridge" plug-in provides a scripting interface to control FSI Viewer using JavaScript. Using the JavaScript bridge you can control viewer functions like "ZoomIn" and "Reset", you can remotely press/release Buttons inside the FSI Viewer and you can zoom to specified image sections directly. Additionally you can implement a JavaScript function receiving FSI event notifications.

Though the integration of the plug-in is very easy, basic knowledge of JavaScript is required to use JSBridge plug-in – especially regarding browser dependent JavaScript implementations.

Implementing the JavaScript Bridge plug-in

1) Add an ID-attribute to your FSI Viewer Object and Embed tag
E.g.

```
<object id="fsiviewer" classid= ...>
<embed NAME="fsiviewer"...>
```

This way you are able to access the viewer object via Javascript.

2) Provide the appropriate action via SetVariable command
To control the FSI Viewer from JavaScript you have to pass the appropriate command by setting the "`FSICMD` " variable of the FSI Viewer movie clip using the `SetVariable()` function.

Example: Button "Zoom In" (MSIE)

```
<input type="button" value="Zoom In"
  onMouseDown="document.all.fsiviewer.SetVariable('FSICMD',
'ZoomIn');"
  onMouseUp  ="document.all.fsiviewer.SetVariable('FSICMD', ''); >
```

Command Parameters

There are 3 different command types you can pass to FSI Viewer / FSI Showcase:

1) Initiate an action directly, e.g. "`ZoomIn`" or "`Reset`"
2) Press and release of FSI Viewer buttons, e.g. "`Button:Mouse_Mode_0`"
3) Zoom to a specified image section, e.g. "`Goto:1,6,0.4,0.2,0.6,0.4`"
   Please see the "Obtaining Image Section Parameters" section for details on how to obtain valid image section parameters.

The following tables list possible values you can pass to FSI Viewer.
Please note that especially regarding the "Button:" command there might be more or not all commands listed available, depending on your FSI Viewer skin and integrated plug-ins.

All command strings have to be provided exactly as stated in the lists below, especially regarding upper-/ lowercase writing and white spaces.

1) Direct Commands

Usage:
`FSIViewer.SetVariable('FSICMD', '[command]');`

To stop the initiated action use:
`FSIViewer.SetVariable('FSICMD', '');`

Command	**Description**
" (empty string)	Stop current action / Release Button
Reset	Reset Viewer
ZoomIn	Start zooming in
ZoomOut	Start zooming out
Up	Start panning upwards
Down	Start panning downwards
Left	Start panning to the left
Right	Start panning to the right
RotateLeft	Start rotating counter-clockwise (Z-axis)
RotateRight	Start rotating clockwise (Z-axis)
NextScene	Start rotating clockwise (Y-axis)
PreviousScene	Start rotating counter-clockwise (Y-axis)
NextSceneSet	Start rotating clockwise (X-axis)
PreviousSceneSet	Start rotating counter-clockwise (X-axis)
__DebugWindow	Show / Hide the debug window (if available)
__InfoWindow	Show / Hide the "about"-window (if available)
Showcase only:	**Description**
PreviousImage	Select previous Image
NextImage	Select next Image
SelectImage	Select image *n* **Important:** Use SetVariable("ImageIndex", *n*) where *n* is [0 .. images] prior to sending this command Alternatively you can select an image relatively from the currently selected image. To select an image relatively you have to set "ImageIndex" to a number prefixed by "+" or "-", e.g. "+1" selects the next image.
SwitchThumbs	Show / Hide Thumb bar

2) Button press / release

Using this command remotely presses/ releases a button inside FSI Viewer / FSI Showcase.
There can not be more than one button pressed at a time. Pressing another button (by script) will automatically release the previous button.

Usage:

```
FSIViewer.SetVariable('FSICMD', 'Button:[ButtonID]');
```

To release the button previously pressed use:

```
FSIViewer.SetVariable('FSICMD', '');
```

Identifying commands of buttons not listed below

There might be some button identifiers that are not listed below, especially buttons integrated by other plug-ins. To obtain command identifiers of these buttons follow the steps below:

1) Activate the debug window of the FSI viewer (set parameter "debug" to "true").
2) Set the "explore" parameter of the jsBridge plug-in to "true".
3) Click the desired button inside FSI Viewer and copy the identifier from the debug window.

ButtonID	**Description**
" (empty string)	Release previously pressed button
Mouse_Mode_*n*	Activate mouse mode *n* . 0 – Zoom 1 – Pan 2 – Rotate 3D 3 – Rotate (Z-axis) There might be more mouse modes available depending on loaded plug-ins
Tool_Reset	"Reset" button, see command "Reset"
Tool_ZoomIn	"Zoom In" button, see command "ZoomIn"
Tool_ZoomOut	"Zoom Out" button, see command "ZoomOut"
Tool_SwitchUI	Show / Hide FSI Viewer user interface
Plugin_SwitchMusic	Toggle music on or off
Plugin_ExtendedNavigation	Toggle display of ExtendedNavigation plug-in
Plugin_HistoryBack	Click "History Back" button of the history plug-in
Plugin_HistoryForward	Click "History Forward" button of the history plug-in

3) Navigating to image sections

To navigate to a specific image section you simply use the "Goto" command followed by a valid image section parameter. Please refer to the chapter "**Debug Mode**" for details on how to obtain image section parameters.

Example: Button "Zoom to image section" (MSIE)

```
<input type="button" value="Zoom to section"
  onClick="document.all.fsiviewer.SetVariable('FSICMD',
'Goto:1,3,0,0,0');">
```

Implementing JavaScript callbacks

Using JavaScript callbacks enables you to react on FSI Viewer / FSI Showcase events by executing JavaScript code. The JSBridge plug-in calls a JavaScript function using the Flash FSCommand architecture providing an event identifier and a parameter.

Please follow the steps below to implement JavaScript callbacks from FSI Viewer / FSI Showcase:

Step 1. Enable JavaScript callbacks by setting the plug-in parameter "Callback" to "true", e.g.

Example: Enabling JavaScript Callbacks

```
<plugin src="jsbridge" callback="true" />
```

Or alternatively via HTTP query:

```
...fsi.swf?cfg=foo&jsbridge_callback=1
```

Step 2. Add a unique identifier to the <object> and the <embed> tag of FSI Viewer / FSI Showcase.

Example: Adding ID / NAME parameters

```
<object id="fsiviewer" classid= ...>
<Param name="movie" value="...>
...
<embed NAME="fsiviewer"...>
...
</object>
```

Step 3. Add a JavaScript function to your HTML page receiving the event notifications via FSCommand like in the example below.

IMPORTANT

The function name has to be exactly like in the example below and it has to start with the ID/NAME parameter specified in **step 2**.

If you do not use the ID/NAME "**fsiviewer**" you have to replace the corresponding script sections in the example below (printed bold) by **your** ID/NAME parameter.

Example: JavaScript Callback Function

```
<script language="javascript" type="text/javascript">

function fsiviewer_DoFSCommand(fsi_event, params) {
  switch(fsi_event){
      case "LoadProgress":
      ...
      break;
  }
}

// Hook for Internet Explorer
if (navigator.appName && navigator.appName.indexOf("Microsoft") != -
1 &&
navigator.userAgent.indexOf("Windows") != -1 &&
navigator.userAgent.indexOf("Windows 3.1") == -1) {
  document.write('<SCRIPT LANGUAGE=VBScript\> \n');
  document.write('on error resume next \n');
  document.write('Sub fsiviewer_FSCommand(ByVal fsi_event, ByVal
params)\n');
  document.write('  call fsiviewer_DoFSCommand(fsi_event,
params)\n');
  document.write('end sub\n');
  document.write('</SCRIPT\> \n');
}
</script>
```

After following the 3 steps described before you will receive the following event notifications:

Event (fsi_event)	**Parameter** (params)	**Description**
InitComplete	---	The viewer has finished parsing the configuration and is about to start loading image data.
ImageInfo	String	Provides concatenated image information. ***See details below this table.**
LoadingComplete	---	The initial load progress of an image completed.
LoadProgress	Number (percent)	Number corresponding to the progress bar inside FSI Viewer.
Press	Button ID	The button "ID" has been pressed.
Release	Button ID	The button "ID" has been released.
ToolTip	String (escaped)	The tool tip "String" is about to be displayed. ***See details below this table.**
View	Image Section	The image section currently displayed. Can be used with the "Goto" command.
ImageUrl	URL template	Provides a template that can be used to build an image URL corresponding to the image being displayed in FSI Viewer. ***See details below this table.**
Action	Action ID	The action "ID" is currently being executed.
Modal	true / false	The viewer enters a passive (true) or an active (false) state.
MouseMode	Mouse Mode ID	The mouse mode "ID" has been activated.
Zoom	Magnification (percent)	The magnification has changed.
ImageSelected	Image index (0..nImages)	A new image has been selected. (**FSI Showcase only**)

Please note that some plug-ins provide additional events. Please refer to the "events" parameter of the corresponding plug-in for details.

The recommended way to react on an event is to extend the `switch...case` block of the `fsiviewer_DoFSCommand(fsi_event, params)` callback function introduced in **Step 3**.

Detailed Event Parameter Description

INFO
The "Info" event parameter provides attributes of the source image currently being displayed. **Example:** `width=8000;height=6000;TilesX=4;TilesY=3;SceneSets=;ImageIndex=4` The attribute "ImageIndex" applies to FSI Showcase only and provides the 0-based index of the image currently selected.

ImageURL
To receive this notification you have to set the "ImageUrls" parameter of the plug-in to "true". (Please refer to "Plug-in Parameters" below for details) The "ImageURL" event parameter provides an URL template that can be used to retrieve rhe image currently displayed in FSI Viewer. **Example:** `http://imageserver.domain/erez3/erez?src=images/zoomimage.fpx &width=`**`[width]`**`&height=`**`[height]`**`&left=0&top=0&right=1&bottom=1 &tmp=`**`[template]`** The URL template can for example be used to set the "src" attribute of an <IMG> tag. Prior to using the template you have to replace the place holders (bold, in brackets) with valid values. The **[template]** place holder applies to eRez imaging servers only.

ToolTip
The "ToolTip" event parameter provides the tool tip text that is about to be displayed. The values depend on the tool tip language selected. You have to use the JavaScript function **unescape()** before displaying this parameter.

JSBridge Plug-in Parameters

Explore	
Description:	Activate debug output of Button IDs
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

If enabled the debug window will display the Button ID of each button pressed inside FSI Viewer. This way you are able to determine Button IDs not listed in this documentation.
Additionally the debug window will output each command received by the JSBridge plug-in.

Callback	
Description:	Enable JavaScript callbacks
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

If enabled the events of FSI Viewer / FSI Showcase will be forwarded to a specific JavaScript function.
Please refer to the section "**Implementing JavaScript Callbacks**" for details

ImageUrls	
Description:	Provide ImageUrl templates with JavaScript callbacks
Syntax:	Boolean
Default:	false
Context:	Plug-in attribute

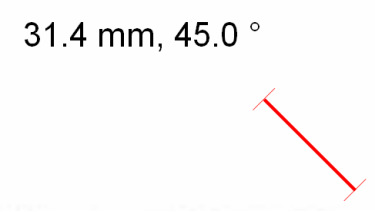
This parameter depends on the "Callback" parameter above. If "Callback" and "ImageUrls" parameter have been set to "true", "ImageUrl" notifications providing image URL templates will be sent to the JavaScript callback function.
Please refer to the "**ImageUrl"** callback identifier for details.

NeptuneLabs FSI Viewer Plug-in
Plug-in Measure

Plug-in Target:
User interface

Plug-in Location:
/plugins/measure.plg

Function:
Provides distance and angle measuring



Syntax:
```
<Plugin src="measure" />
```

This plug-in can be optionally obtained.
Using the plug-in without a valid licence key activates the evaluation mode.

The Measure plug-in provides an additional mouse mode which enables the user to measure distances and angles inside the FSI Viewer by clicking & dragging.

Pressing **SHIFT** locks the angle to 45 degree steps.
Pressing **CTRL** moves the measurement line.

Mouse Mode

The mouse mode id for the measuring mode is 100.
You can use this id to set the InitialMouseMode parameter of the FSI Viewer.
e.g.

```
<InitialMouseMode value="100" />
```

or

```
?cfg=foo&InitialMouseMode=100
```

Setting up the Plug-in

The Measure Plug-in requires the real width of the entire image to enable distance measuring.

Example 1:
If you already know the entire width of the image simply add the "ImageWidth" and the "Suffix" parameter to the plug-in tag:

If the width of the entire image is 120.5 inches the corresponding plug-in tag looks as follows:

```
<plugin src="measure" ImageWidth="120.5" Suffix=" inches" />
```

Please note:
You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "`measure_`".

E.g.:

```
?cfg=image&Measure_ImageWidth=120.5&Measure_Suffix=%20inches
```

Example 2:
If you know the width of a part of the image, but you do not know the width of the entire image, you should follow the steps below:

1. Add the measure plug-in without any parameters.
   `<Plugin src="measure" />`
2. Open the image inside the FSI Viewer and choose the "Measure" mouse mode.
3. Measure the part of the image you know the real width of.
   (The Measure plug-in will display a decimal number *n* between 0 and 1)
4. The value *w* for the ImageWidth parameter is:

   w = *RealWidth* / *n*.

- The Measure plug-in displays *n=0.91388889.*
- The real length *L* of the engine is 214.5 mm.
- ImageWidth *w = L / n = 214.5 / 0.91388889 =* ***234.7112459***

The corresponding plug-in tag is the following:

```
<plugin src="measure" ImageWidth="234.7112459" Suffix=" mm" />
```

Please note:
You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "`measure_`".

E.g.:

```
?cfg=engine&Measure_ImageWidth=234.7112459&Measure_Suffix=%20mm
```

Accuracy

The plug-in does not support any perspective correction.
This means that measuring scanned maps, diagrams will be accurate, measuring 3 dimensional objects is less accurate.

Plug-in Parameters

The following attributes can be assigned to the <plugin> tag.
You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "`measure_`" with the parameter name, e.g. "`measure_ImageWidth`"

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Specifies the space in pixel left of the menu button of the plug-in.

ImageWidth	
Description:	Real width of the entire image
Syntax:	Number
Default:	100
Context:	Plug-in attribute

Defines the real width of the entire image (e.g. 120 inches).
Please see the explanation in the previous section.

Prefix	
Description:	Defines a prefix for the length value
Syntax:	String
Default:	---
Context:	Plug-in attribute

Defines the prefix for the length value e.g. "length: ".

Suffix	
Description:	Defines a suffix for the length value
Syntax:	String
Default:	---
Context:	Plug-in attribute

Defines the suffix for the length value e.g. " inches". Usually this is a length unit.

Decimals	
Description:	Number of decimals of the length value
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Defines the number of decimals of the length value.

ShowText	
Description:	Display the current measurement string
Syntax:	Bool
Default:	True
Context:	Plug-in attribute

Hides the text displaying the current measurement if set to “false”.

ShowAngle	
Description:	Display current angle
Syntax:	Bool
Default:	false
Context:	Plug-in attribute

If set to true the plug-in appends a text representing the current angle in degrees to the text display. E.g. "120.2 inches, 43.2°".

LineColor	
Description:	Defines the color of the measure lines
Syntax:	HexColor
Default:	FF0000
Context:	Plug-in attribute

6-digit hexadecimal color value defining the color of the measure lines.

TextColor	
Description:	Defines the color of the text
Syntax:	HexColor
Default:	000000
Context:	Plug-in attribute

6-digit hexadecimal color value defining the color of the text.

BGColor	
Description:	Defines the color of the text background
Syntax:	HexColor
Default:	FFFFFF
Context:	Plug-in attribute

6-digit hexadecimal color value defining the background color of the text.

BGAlpha	
Description:	Defines the opacity of the text background
Syntax:	Number
Default:	80
Context:	Plug-in attribute

Defines the opacity of the text background (0...100).

Events	
Description:	Fire plug-in events
Syntax:	Boolean
Default:	False
Context:	Plug-in attribute
Version:	3.2.5 and above

Specifies whether the plug-in fires plug-in events. Plug-in events can be handled when using JSBridge plug-in or FSI Viewer Component.

The following events apply to this plug-in:

Event ID	Parameter	Description
onMeasureEnable	Boolean	The measure tool gets enabled or disabled.
onMeasureStart	String	The user starts measuring. The parameter provides the current distance and angle as displayed in the measure tool.
onMeasure	String	The user drags the measure tool. The parameter provides the current distance and angle as displayed in the measure tool.
onMeasureStop	String	The user stopped measuring. The parameter provides the current distance and angle as displayed in the measure tool.

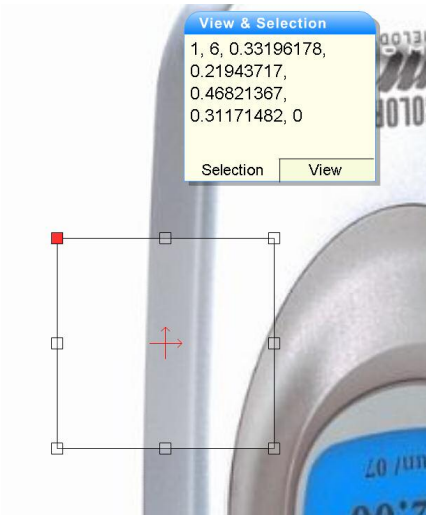
NeptuneLabs FSI Viewer Plug-in
Plug-in SelectFrame

Plug-in Target:
User interface

Plug-in Location:
/plugins/selectframe.plg

Function:
Provides a resizable select frame

Syntax:
`<Plugin src="selectframe" />`



This plug-in adds a resizable select frame into FSI Viewer.

Pressing **SHIFT** locks the aspect ratio of the selection.
Pressing **CTRL** resizes the selection relative to the center
Pressing **ALT** rotates the selection
Pressing **ALT + SHIFT** rotates the selection in steps of 5 degrees

Plug-in Parameters

The following attributes can be assigned to the <plugin> tag.
You might as well define all plug-in parameters via query or inside the <options> group by using the prefix "`measure_`" with the parameter name, e.g. "`selectframe_window`"

MenuOffset	
Description:	Indentation of the menu button
Syntax:	Number
Default:	0
Context:	Plug-in attribute

Specifies the space in pixel left of the menu button of the plug-in.

Visible	
Description:	Show or hide the plug-in on startup
Syntax:	Boolean
Default:	False
Context:	Plug-in attribute

Show or hide the select box on startup.

Window	
Description:	Show or hide the output window
Syntax:	Boolean
Default:	True
Context:	Plug-in attribute

Show or hide the output window of the plug-in.

Rotate	
Description:	Enable rotation of the selection
Syntax:	Boolean
Default:	True
Context:	Plug-in attribute

Enable rotating the selection by pressing **ALT** key or **ALT + SHIFT** keys and dragging the handles of the select box.

MaskColor	
Description:	Color of the mask
Syntax:	HexColor
Default:	FFFFFF
Context:	Plug-in attribute

Define a 6-digit hexadecimal color value for the mask of the select box.

MaskAlpha	
Description:	Opacity of the mask
Syntax:	Number
Default:	75
Context:	Plug-in attribute

Defines the opacity of the mask of the select box (0 = invisible).

AspectRatio	
Description:	Aspect ratio of the selection
Syntax:	String
Default:	---
Context:	Plug-in attribute

Defines an aspect ratio for the selection, e.g. "320:200".
Use SHIFT to lock or unlock the aspect ratio when resizing the selection.

InvertShift	
Description:	Inverts the function of the SHIFT key
Syntax:	Boolean
Default:	False
Context:	Plug-in attribute

You can use this parameter to invert the function of the SHIFT key while dragging.

Callback	
Description:	JavaScript callback function
Syntax:	String
Default:	---
Context:	Plug-in attribute

Specify the name of a custom JavaScript function to call each time the selection changes. The arguments passed to this function are:

- SceneSet, Scene, Left, Top, Right, Bottom, Rotation, Current View

Example: JavaScript callback function

```
function onSelect(set, scene, l, t, r, b, rot, strView){
  doSomething();
}
```

Additionally you can as use the template "$$`selection`$$" for the → **CustomButton** plug-in.

Events	
Description:	Fire plug-in events
Syntax:	Boolean
Default:	False
Context:	Plug-in attribute
Version:	3.2.5 and above

Specifies whether the plug-in fires plug-in events. Plug-in events can be handled when using JSBridge plug-in or FSI Component.

The following events apply to this plug-in:

Event ID	Parameter	Description
onSelectFrameEnable	Boolean	The select has been enabled or disabled.
onSelectFrameStart	---	The user starts selecting.
onSelectFrame	String	The user modifies the select frame. The parameter provides a string representing the current selection.
onSelectFrameStop	String	The user stopped modifying the selection. The parameter provides a string representing the current selection.

Error Messages

Please make sure to enable the debug mode when encountering errors !

Usually it is easy to locate the reason for an error by examining the output in the debug window or the step of the initialization process where the FSI Viewer stops.

Warning: NeptuneLabs ZoomCache required for Flash 5 Plug-in

You may ignore this warning if you plan to use the FSI Viewer with Flash plug-ins version 6 and above only.
To use FSI Viewer with Flash plug-ins from version 5 NeptuneLabs ZoomCache is required.

Warning: ViewerWidth and ViewerHeight required

You may ignore this warning if you plan to use the FSI Viewer with Flash plug-ins version 6 and above only.
If you plan to use the FSI Viewer with Flash plug-ins prior to version 6 you have to adjust or add the FSI parameters ViewerWidth and ViewerHeight according to the dimension of the <object> and <embed> tag.

Error: FSI Viewer does not show up at all

Please check the parameters of the <object> and <embed> tag.
Make sure that the HTML code:

<PARAM NAME="movie" VALUE="**[URL and parameter]** ">

contains the correct URL to the FSI Viewer file "fsi.swf".

Please make sure to have Flash plug-in 5 or above installed.

Please make sure that all required files have been transferred in binary mode to your web server.
If not, adjust the corresponding configuration options of your FTP client accordingly and upload all FSI Viewer files once again.

Error: Loading _default.fsi...failed

Please make sure that the file "_default.fsi" resides in the setup directory of your FSI Viewer.
Check the _default.fsi file for XML syntax errors.

Error: Loading *.fsi...failed
or
Error: FPXSrc undefined

Please make sure that the configuration file defined by the CFG parameter is valid.
Please make sure that the FSIBase parameter in the _default.fsi file points to the correct directory.
Please check the XML syntax of the corresponding FSI-file.

Error: The Viewer stops displaying the message "Loading Skin and FSI Module:"

Please make sure that all files have been uploaded to the location provided when ordering your FSI product.

Error: The Viewer stops displaying the message "Opening FPX: [...]"

Please check the given path to the source image file on the imaging server.
Adjust the parameters FPXSrc or FPXBase accordingly.

Error: Viewer stops displaying the message "Loading Plug-in ..."

Please check the "**Src**" parameter of the specified plug-in.
Make sure that the corresponding plug-in file exists in the "/plugins" subdirectory of your setup directory of your web server.

Error: The image appears entirely black or distorted

This might happen due to an error during the conversion of your image to the FPX format.
Please make sure that the source image has not been saved in grayscale mode and that it does not contain multiple alpha channels.
Be sure to save the image without FPX data compression.
Please try converting your image to RGB color mode and remove all alpha channels prior to FPX conversion.

Error: The Viewer displays multiple images at a time

You might be using an FPX containing multiple images without defining the parameters TilesX and TilesY or these parameters have been defined incorrectly.
Please check the output of the Debug Window for the scene sets setting.

Error: The Viewer displays a clipped image

You might have defined the parameters TilesX and TilesY for a 2D image.
Or
The parameters ViewerWidth and ViewerHeight do not match the corresponding settings in the <object> or <embed> tag using Flash plug-in version 5.

Error: Keyboard commands not working

Using the keyboard commands require the FSI Viewer to have the input focus.
Click anywhere inside the Viewer to pass the focus to the FSI Viewer.

Appendix

Example of a _default.fsi file

```
<fsi_parameter>
   <!-- This file contains default parameters.
   All parameters will be overwritten by additional .fsi files or query
   parameters. -->

   <FPX>
      <ServerType value="eRez" />
      <Base value="http://erez.neptunelabs.com/erez3/erez?src=/" />
   </FPX>

   <Plugins>
      <Plugins src="mousemodes" />
   </Plugins>

   <Options>
      <FSIBase value="../fsi/" />

      <ScenePreload value="true" />
      <MenuAlign value="BR" />
      <Animation value="BEST" />

      <ZoomCache      value="http://zoomcache.provider.com" />
      <ZoomCacheID    value="my_zoomcache_id" />
      <ZoomCache5Only value="true" />
   </Options>
</fsi_parameter>
```

Example of an image specific FSI configuration file

```
<fsi_parameter>

   <!-- required parameters for Flash 5 plugin -->
   <Viewer>
      <Width     value="330" />
      <Height    value="338" />
   </Viewer>

   <FPX>
      <!-- Src will be prefixed by FPXBase of _default.fsi -->
      <Src       value="3d_object.fpx" />
      <Width     value="8128" />
      <Height    value="9168" />
      <TilesX    value="4" />
      <TilesY    value="3" />
   </FPX>

   <Options>
      <InitialAction value="NextScene" />
      <InitialActionDelay value="3" />
      <MenuAlign value="TL" />
      <NoNav value="true" />
   </Options>

</fsi_parameter>
```

Credits

Special thanks go to the following persons for their valuable support such as providing translations, sharing their ideas regarding improvements and beta testing:

Bora Alioglu
Oscar Applegren
Sven-Erik Andersen
Radoslav Bielik
Maciej Chojnacki
Jacky Choy
Petr Cihar
Marc Delisle
Daniel Hinostroza
Yukihiro Kawada
Kristian Ottosen
Ivo Teel
Andri Viiand

Index

Absolute Addressing 29
Automation 25, 55
Comments 22
Compatibility and Requirements 10
Debug Mode 53
Embed Tag 18
Error Messages 142
Flash 5 32
FSI Showcase 61
FSI Showcase Parameters 68
FSI Viewer 20
Installation 15
Keyboard 14
Menu bar 11
Mouse cursor 38
Mouse Modes 12
Navigation 11
Object Tag 18
Parameter
  Animation 39
  AnimationSpeed 39
  Cfg 27
  ConcurrentRequests 45
  Debug 14, 30
  DefaultCFG 27
  DetailBuffer 33
  Effects 40
  ExtendedViewport 39
  FPXBase 31
  FPXServerTemplate 28
  FPXServerType 28
  FPXSrc 28, 143
  FPXTilePaddingX 51
  FPXTilePaddingY 51
  FSIBase 31
  HandCursor 38
  Height 29, 32
  HelpURL 37
  HelpUrlTarget 37
  HideUI 40
  IgnoreQuery 44
  InitalView 35
  InitialAction 36
  InitialActionDelay 36
  InitialActionPersistent 37
  InitialMouseMode 37
  InitialViewPersistent 36
  Intro 30
  Language 33
  MaxZoomLevel 44
  MenuAlign 33
  NavEffects 43
  NavFrame 34
  NavFrameColor 34
  NavHeight 34
  NavMaskAlpha 35
  NavMaskColor 35
  NavWidth 34
  NoImageBlend 40
  NoNav 34
  NoSceneAnimation 51
  NoSceneLoop 52
  NoSetLoop 51
  SceneEffects 43
  ScenePreload 51
  SceneSets 50, 51, 144
  ServerTemplate 28
  ServerType 28
  Skin 30
  Src 28, 58
  TileEffects 43
  TileSizeX 45
  TilesX 49, 144
  TilesY 49, 144
  UISwitchable 40
  Width 29, 32
  ZoomCache 43
  ZoomCache5Only 44
  ZoomCacheID 44
  ZoomLevelFrequency 46
  ZoomLimit 38
  ZoomLimitMin 38
Plugin-Reference 86
Plugins
  Clock Progress 106
  History 92
  HotSpot 113
  JSBrdige 123
  Magnifier 111
  Measure 132
  Mouse Mode Select 88
  Mouse Modes 86
  Mousewheel 103
  Music 109
  Nav Extension 100
  Resize 94, 96
  SelectFrame 138
  Software Cursor 102
  Zoom Meter 90
Plug-ins 86

Relative Addressing 29
Rotation..... 12, 14
Scenes.....50, 51, 144
Survey window 13, 33, 34, 35
Thumbnail Effects.....73
ViewerBorder.....32

FSI Viewer – FSI Showcase
Flash based Single Source Image Viewer

NeptuneLabs GbR
P.O.B. 1207
32818 Blomberg
Germany

Fon: +49 (0) 5236-888559
Fax: +49 (0) 5236-888560
eMail: info@neptunelabs.com
WWW: www.neptunelabs.com

No part of this manual, including the software described in it, may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means, except documentation kept by the purchaser for backup purposes, without the express written permission of NeptuneLabs.

Specifications and information contained in this manual are furnished for informational use only and are subject to change at any time without notice, and should not be construed as a commitment by NeptuneLabs. NeptuneLabs assumes no responsibility or liability for any errors or inaccuracies in this manual, including the software described in it.

Copyright 2002-2005 NeptuneLabs GbR. All rights reserved.